

**DISEÑO E IMPLEMENTACIÓN DE UNA APPS WEB QUE MUESTRE LA  
DISPONIBILIDAD EN LOS PARQUEADEROS DE LA CIUDAD DE MONTERÍA  
DEL DEPARTAMENTO DE CÓRDOBA.**

**JEINA JANITH GÓMEZ ROMERO**

**ADELA MERCEDES BARRIOS SANCHEZ**

**TRABAJO DE GRADO**

**UNIVERSIDAD DE CÓRDOBA**

**FACULTAD DE INGENIERÍAS**

**INGENIERÍA DE SISTEMAS**

**MONTERÍA**

**2015**

**DISEÑO E IMPLEMENTACIÓN DE UNA APPS WEB QUE MUESTRE LA  
DISPONIBILIDAD EN LOS PARQUEADEROS DE LA CIUDAD DE MONTERÍA  
DEL DEPARTAMENTO DE CÓRDOBA.**

**JEINA JANITH GÓMEZ ROMERO**

**ADELA MERCEDES BARRIOS SANCHEZ**

**ASESOR:**

**Mg. HAROLD BULA HERAZO**

**TRABAJO DE GRADO PARA OBTENER EL TITULO DE:  
INGENIERO DE SISTEMAS**

**UNIVERSIDAD DE CÓRDOBA**

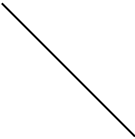
**FACULTAD DE INGENIERÍAS**

**INGENIERÍA DE SISTEMAS**

**MONTERÍA**

**2015**

**NOTA DE ACEPTACION**



---

---

---

---

---

---

---

*Director*

---

*Firma Del Jurado 1*

---

*Firma Del Jurado 2*

Montería, Córdoba (Día \_\_\_\_ Mes \_\_\_\_ Año \_\_\_\_)

## **DEDICATORIA**

*Dedico este trabajo principalmente a dios por ser el artífice, creador y responsable de todos mis éxitos, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el proceso de estudio. a mis padres Inés verónica Sánchez Agámez y Ronald Barrios Reyes, por su incondicional apoyo durante esta travesía, a mis hermanos Ronald Amaury y Yiseth Paola Barrios Sánchez por brindarme todo el cariño y apoyo necesario para culminar este objetivo, a mi novio Ferney Argumedo Solera por su acompañamiento y apoyo incondicional, a mis abuelos(as), tías (os), primos(as), a todos mis amigos y compañeros de estudio en especial a “el combo”, a mi amiga y compañera de tesis Jeina Gómez Romero, a otros amigos a lo largo de la carrera como lo es Andy Maluth Restrepo, Miguel Barroso y a todas aquellas personas que de alguna forma han hecho parte de esta etapa de crecimiento personal.*

**ADELA MERCEDES BARRIOS SÁNCHEZ**

*Dedico este nuevo logro en mi vida primeramente, dándole infinitas gracias a Dios por ser mi guía, llenarme de mucha fe, fortaleza, paciencia y sabiduría para afrontar cada reto en este camino.*

*Infinitas gracias doy a mis padres Jimmy Gómez Valverde y Miriam Romero Fabra que cada día me brindaron su paciencia, motivación, sacrificio, entrega para formarme como una persona de bien, integra y llena de valores, gracias a ustedes y a mi hermana Jania Gómez Romero por apoyarme y ser parte de este nuevo éxito alcanzado.*

*Dedico este logro a mis demás familiares y en especial a una persona que me motivó mucho para seguir adelante con mis sueños, gracias por su apoyo incondicional y por influir en mi vida para alcanzar todos los objetivos propuestos, es también para ustedes este triunfo en agradecimiento por todo su amor.*

*Gracias a esas personas que siempre estuvieron presente en los momentos difíciles, para darme su mano y enseñarme a crecer, con cariño les dedico a ustedes: primeramente a mi compañera, amiga y hermana Erika Castro Góez, gracias por apoyarme y estar en todo momento conmigo, a mis compañeros y amigos Adela Barrios Sánchez, Andy Restrepo López y Miguel Barroso Severiche, quienes me acompañaron en tantos desvelos y sufrimientos a los largo de mi formación profesional, infinitas gracias.*

**JEINA JANITH GÓMEZ ROMERO**

## **AGRADECIMIENTOS**

*Queremos agradecer a la Universidad de Córdoba por brindarnos la oportunidad de ser parte de su alma mater y convertirnos en profesionales, y a todos los docentes que hicieron parte de nuestro proceso académico.*

*A nuestro tutor y asesor Harold Bula Herazo, por brindarnos sus asesorías y conocimientos durante este largo proceso, infinitas gracias por ser parte de nuestro logro.*

*A cada uno de nuestros familiares y amigos por todo el apoyo recibido durante nuestro proceso de formación, sin ellos el camino recorrido y los logros alcanzados hubiesen sido más difíciles de superar.*

*A nuestros compañeros, como olvidar a quienes sin pedir nada a cambio, siempre estuvieron dispuestos a brindarnos toda su colaboración, aunque es poco para lo mucho que nos ofrecieron, nuestra inmensa gratitud a ustedes.*

## **TITULO**

**DISEÑO E IMPLEMENTACIÓN DE UNA APPS WEB QUE MUESTRE LA  
DISPONIBILIDAD EN LOS PARQUEADEROS DE LA CIUDAD DE MONTERÍA  
DEL DEPARTAMENTO DE CÓRDOBA.**

## Contenido

<b>1. OBJETIVOS.....</b>	<b>16</b>
<b>1.1. OBJETIVO GENERAL.....</b>	<b>16</b>
<b>1.2. OBJETIVOS ESPECÍFICOS.....</b>	<b>16</b>
<b>2. INTRODUCCIÓN .....</b>	<b>17</b>
<b>3. MARCO TEÓRICO .....</b>	<b>23</b>
<b>3.1. SISTEMA DE INFORMACIÓN GEOGRÁFICA .....</b>	<b>23</b>
<b>3.3. APPS WEB.....</b>	<b>33</b>
<b>3.4. BASES DE DATOS .....</b>	<b>35</b>
<b>3.5. SENSORES .....</b>	<b>38</b>
<b>3.6. ARDUINO .....</b>	<b>41</b>
<b>4. MARCO CONCEPTUAL .....</b>	<b>44</b>
<b>4.1. GPS .....</b>	<b>44</b>
<b>4.2. HTML .....</b>	<b>44</b>
<b>4.3. CSS3.....</b>	<b>44</b>
<b>4.4. LENGUAJE DE PROGRAMACIÓN PHP .....</b>	<b>45</b>
<b>4.5. JAVA .....</b>	<b>45</b>
<b>4.6. JAVASCRIPT.....</b>	<b>45</b>
<b>4.7. INTERFAZ .....</b>	<b>46</b>
<b>4.8. C++.....</b>	<b>46</b>



<b>4.9. ICSP .....</b>	<b>46</b>
<b>4.10. PWM.....</b>	<b>46</b>
<b>4.11. LENGUAJE PROCESSING .....</b>	<b>47</b>
<b>4.12. WIRING.....</b>	<b>47</b>
<b>5. METODOLOGIA .....</b>	<b>48</b>
<b>5.1. FASES DEL PROYECTO.....</b>	<b>49</b>
<b>5.2. METODOLOGÍA DEL DESARROLLO DEL PRODUCTO .....</b>	<b>51</b>
<b>6. DESARROLLO .....</b>	<b>54</b>
<b>6.1. ARQUITECTURA DEL SISTEMA .....</b>	<b>54</b>
<b>6.2. DISEÑO DEL SISTEMA.....</b>	<b>58</b>
<b>7. CONCLUSIONES.....</b>	<b>70</b>
<b>BIBLIOGRAFÍA.....</b>	<b>71</b>
<b>ANEXO .....</b>	<b>75</b>

## LISTA DE ILUSTRACIONES

Ilustración 1: Estructura de un SIG, Fuente: <i>laciudadvivaoirg</i> .....	24
Ilustración 2: Representativa de Apps Web, Fuente: <a href="http://rho-design.com/project/creacion-de-apps-y-web-apps/">http://rho-design.com/project/creacion-de-apps-y-web-apps/</a> .....	35
Ilustración 3: Sensores Fotoeléctricos, Fuente: <a href="http://mco-s2-p.mlstatic.com/barrera-fotoelectronica-perimetral-doble-lente-5870-MCO5007799911_092013-O.jpg">http://mco-s2-p.mlstatic.com/barrera-fotoelectronica-perimetral-doble-lente-5870-MCO5007799911_092013-O.jpg</a> .....	41
Ilustración 4: Tipos de Arduinos. (Google, s.f.), Fuente: <a href="http://todotecnologia-es.blogspot.com.co/2014/12/que-modelo-arduino-comprar.html">http://todotecnologia-es.blogspot.com.co/2014/12/que-modelo-arduino-comprar.html</a> .....	41
Ilustración 5: Modelo Cascada, Fuente: Diseño propio. ....	51
Ilustración 6: Arquitectura del sistema MVC y las herramientas utilizadas en cada una de las capas del modelo. ....	56
Ilustración 7: Arquitectura del dispositivo y el sistema, Fuente: Diseño Propio .....	57
Ilustración 8: Diseño del sistema, Fuente: Diseño propio .....	58
Ilustración 9: Interfaz de Inicio de la Apps Web para el administrador de la aplicación. ....	59
Ilustración 10: Primera Interfaz del administrador de la aplicación .....	60
Ilustración 11: Interfaz del administrador cuando agrega el marcador del nuevo parqueadero. ....	60
Ilustración 12: Interfaz del formulario para insertar los datos del administrador del parqueadero nuevo.....	61
Ilustración 13: Interfaz del administrador de la Apps Web cuando edita un parqueadero. ....	62
Ilustración 14: Interfaz del administrador del parqueadero cuando ingresa sus datos..	63

Ilustración 15: Primera Interfaz del administrador del parqueadero cuando inicia sesión. ....	63
Ilustración 16: Interfaz del formulario para ingresar un nuevo portero. ....	64
Ilustración 17: Interfaz del portero del parqueadero cuando inicia sesión. ....	65
Ilustración 18: Primera Interfaz del portero del parqueaderos después de iniciar sesión. ....	65
Ilustración 19: Interfaz del formulario que llena el portero del parqueadero para un nuevo parqueo. ....	66
Ilustración 20: Interfaz de Inicio de la Apps Web .....	67
Ilustración 21: Interfaz cuando el usuario escoge un parqueadero y desea ver su disponibilidad .....	68
Ilustración 22: Interfaz cuando el usuario desea ver más información sobre un parqueadero en específico. ....	69
Ilustración 23: Diagrama de subsistemas.....	82
Ilustración 24: Diagrama casos de uso 01 Control Acceso.....	83
Ilustración 25: Diagrama casos de uso 02 Sistema de GPS.....	84
Ilustración 26: Diagrama de caso de uso 07 Administración de parqueadero. ....	85
Ilustración 27: Diagrama de caso de uso 08: Navegación. ....	86
Ilustración 28: Diagrama de caso de uso 09: Búsqueda. ....	87
Ilustración 29: Diagrama de caso de uso 11: Administración de portero. ....	88
Ilustración 30: Diagrama de caso de uso 12: Estadísticas del sistema.....	89
Ilustración 31: Diagrama de Secuencia 01 Control de acceso. ....	97
Ilustración 32: Diagrama de Secuencia 02 Sistema de GPS.....	98

Ilustración 33: Diagrama de Secuencia 03 Administración de parqueaderos opción agregar. ....	99
Ilustración 34: Diagrama de Secuencia 03 Administración de productos opción modificar.....	100
Ilustración 35: Diagrama de Secuencia 03 Administración de parqueaderos opción eliminar.....	101
Ilustración 36: Diagrama de Secuencia 04 Navegación del Administrador de la Apps Web.....	102
Ilustración 37: Diagrama de Secuencia 04 Navegación del Administrador del parqueadero. ....	103
Ilustración 38: Diagrama de Secuencia 04 Navegación de portero. ....	104
Ilustración 39: Diagrama de Secuencia 04 Navegación de Usuario. ....	105
Ilustración 40: Diagrama de Secuencia 05 Búsqueda. ....	106
Ilustración 41: Diagrama de Secuencia 07 Administración de portero opción agregar. .....	107
Ilustración 42: Diagrama de Secuencia 07 Administración de porteros opción modificar.....	108
Ilustración 43: Diagrama de Secuencia 07 Administración de portero opción eliminar. .....	109
Ilustración 44: Diagrama de Secuencia 08 Estadísticas del sistema. ....	110
Ilustración 45: Diagrama de Estado 01 Control de acceso. ....	111
Ilustración 46: Diagrama de Estado 02 Sistema de GPS. ....	112
Ilustración 47: Diagrama de Estado 03 Administración de Parqueaderos.....	112
Ilustración 48: Diagrama de Estado 04 Navegación del Adm. De la Apps web. ....	113

Ilustración 49: Diagrama de Estado 04 Navegación del Admon. Del parqueadero. ..	114
Ilustración 50: Diagrama de Estado 04 Navegación del Portero Del parqueadero.....	114
Ilustración 51: Diagrama de Estado 04 Navegación del Usuario .....	115
Ilustración 52: Diagrama de Estado 05 Búsqueda. ....	116
Ilustración 53: Diagrama de Estado 07 Administración de Portero .....	117
Ilustración 54: Diagrama de Estado 08 Estadísticas del sistema .....	118
Ilustración 55: Diagrama de Actividad 01 Control de acceso. ....	119
Ilustración 56: Diagrama de Actividad 02 Sistema GPS .....	120
Ilustración 57: Diagrama de Actividad 03 Administración de parqueaderos opción agregar. ....	121
Ilustración 58: Diagrama de Actividad 03 Administración de parqueaderos opción modificar.....	121
Ilustración 59: Diagrama de Actividad 03 Administración de parqueaderos opción eliminar.....	122
Ilustración 60: Diagrama de Actividad 04 Navegación del Administrador de la Apps Web.....	123
Ilustración 61: Diagrama de Actividad 04 Navegación del Administrador del parqueadero. ....	123
Ilustración 62: Diagrama de Actividad 04 Navegación del portero.....	124
Ilustración 63: Diagrama de Actividad 04 Navegación del Usuario .....	124
Ilustración 64: Diagrama de Actividad 03 Búsqueda. ....	125
Ilustración 65: Diagrama de Actividad 04 Administración de usuario opción actualizar. .....	126

Ilustración 66: Diagrama de Actividad 04 Administración de usuario opción actualizar.	
.....	126
Ilustración 67: Diagrama de Clase. ....	128
Ilustración 68: Diagrama Entidad relación .....	131

## LISTA DE TABLAS

Tabla 1: Fortalezas de MySQL .....	38
Tabla 2: Tipos de Sensores (Balcells J., s.f.) .....	39
Tabla 3: Fases de investigación .....	50
Tabla 4: Etapas de desarrollo del producto .....	53
Tabla 5: Objetivos del sistema. ....	77
Tabla 6: Requisitos de información control de acceso. ....	78
Tabla 7: Requisitos de información Sistema de GPS. ....	78
Tabla 8: Requisitos de información administración de productos. ....	79
Tabla 9: Requisitos de información de navegación. ....	79
Tabla 10: Requisitos de información Búsqueda .....	80
Tabla 11: Requisitos de información administración de porteros.....	80
Tabla 12: Requisitos de información de estadísticas del sistema. ....	81
Tabla 13: Documentación del caso de uso del control de acceso. ....	90
Tabla 14: Documentación del caso de uso de sistema de GPS. ....	91
Tabla 15: Documentación del caso de uso de administración de parqueaderos. ....	91
Tabla 16: Documentación del caso de uso de Navegación. ....	92
Tabla 17: Documentación del caso de uso de búsqueda. ....	92
Tabla 18: Documentación del caso de Administración de portero. ....	93
Tabla 19: Documentación del caso de Estadísticas del sistema.....	94
Tabla 20: Documentación de clase Parqueadero .....	128
Tabla 21: Documentación de clase usuarios. ....	129
Tabla 22: Documentación de clase Parqueo. ....	129

## **1. OBJETIVOS**

### **1.1. OBJETIVO GENERAL**

Diseñar e implementar una Apps web que muestre la disponibilidad en los parqueaderos de la ciudad de montería del departamento de córdoba.

### **1.2. OBJETIVOS ESPECÍFICOS**

- Identificar requerimientos y establecer mayor comodidad para localizar un parqueadero que tenga un lugar disponible.
- Explorar conceptos, tecnologías y aplicaciones existentes para soportar servicios de información y publicidad en dispositivos móviles.
- Diseñar en los parqueaderos un sistema digital que detecte el flujo de vehículos, utilizando sensores fotoeléctricos de entrada y salida, que controle el acceso al parqueadero, prohibiendo el paso de vehículos cuando éste se encuentre lleno.
- Construir un prototipo web que demuestre la funcionalidad de la aplicación propuesta.
- Desarrollar pruebas de funcionamiento para determinar la aceptación del aplicativo entre los usuarios de los parqueaderos de la ciudad de Montería.



## 2. INTRODUCCIÓN

Los avances tecnológicos han mejorado la calidad de vida del hombre, generando un gran impacto en el modo de vivir de la sociedad y se han convertido en una herramienta fundamental que brindan la oportunidad de hacer de las actividades diarias, algo más prácticas, eficaces y sencillas.

Hoy día Google Inc. ha realizado importantes aportes a estos avances; uno de los más relevantes para la realización de este proyecto es la API de Google Maps, la cual es una interfaz que permite dibujar una posición en un mapa y hacer consultas geográficas sobre sitios específicos, estos mapas se puede visualizar desde un sitio web (Khaled Abou & Joshua, 2013).

Esta nueva tecnología de localización está transformando el mundo, permitiendo a millones de usuarios realizar actividades que anteriormente parecían imposibles; esta nueva experiencia ha mejorado el mundo sobrepasando fronteras que muchas veces se vuelven obstáculos en el entorno en el que se vive.

La unión de la tecnología API de Google Maps con dispositivos electrónicos hace que surjan nuevos proyectos innovadores que facilitan los procesos; Entre los dispositivos electrónicos que más se utilizan para llevar a cabo estos proyectos se encuentra la plataforma de hardware libre basada en una placa con microcontroladores y un entorno de desarrollo llamada ARDUINO, esta placa permite programarse según la necesidad del proyecto, además admite la conexión con otro tipo de dispositivos electrónicos que envían, guardan o muestran la información. La SHIELD ETHERNET es uno de estos dispositivos que permite conectarse con el Arduino y ofrecerle acceso a internet.

Para la ejecución del proyecto Easy Parking se utilizaron SENSORES FOTOELECTRICOS, los cuales usan un haz de luz para detectar la presencia o ausencia de

un objeto, en este caso detectara la presencia o ausencia de un vehículo permitiendo conocer la disponibilidad de puestos que hay en un parqueadero.

Actualmente, en la ciudad de Montería, se observa el aumento del flujo vehicular generando así la escasez de puestos en parqueaderos, lo que se ha convertido en un problema de movilidad para los automovilistas y la población en general, puesto que se presenta congestión, contaminación, reducción de espacios y pérdida de tiempo a la hora de buscar un lugar para estacionar el vehículo.

El problema de Movilidad en esta ciudad es una realidad tangible que se vive día a día, debido al crecimiento del parque automotor y con esto la falta de espacio donde estacionar dicho conglomerado vehicular; razón por la cual, el sector público y privado se ha visto en la necesidad de crear un conjunto de acciones que solucionen esta problemática, dentro de las cuales se contemplan medidas eficaces frente a la falta de parqueaderos y mejoramiento en la calidad del servicio que prestan los mismos.

Estas medidas buscan básicamente contribuir con soluciones prácticas e innovadoras a los problemas de movilidad, puesto que, cada día miles de vehículos transitan por la ciudad y no siempre consiguen un lugar de estacionamiento cercano a sus trabajos, lo cual hace colapsar en poco tiempo las vías principales del centro de la ciudad y los barrios aledaños a él.

Las nuevas construcciones inmobiliarias que no siempre cuentan con estacionamiento para todos los departamentos; la falta de creación de nuevas plazas de estacionamiento; los contenedores de basura; y la extensión de las bici sendas son múltiples factores que hacen que parquear el coche en la calle sea una ardua tarea cotidiana.

Encontrar dónde parquear en la ciudad de Montería puede sacarnos de nuestras casillas; es más, se calcula que el 20% del tráfico diario que circula en los centros urbanos

se debe a conductores que buscan donde estacionar, lo que se traduce en atascos continuos y emisiones a la atmósfera de CO<sub>2</sub> que se podrían evitar. Aun cuando el centro de la ciudad ofrece sitios de parqueaderos, éstos suelen estar completos o tienen lugares limitados. También se puede ver que es difícil encontrar un lugar para estacionar en las calles del centro de la ciudad.

Es importante reconocer que la cantidad de parqueaderos que hay en la ciudad no son suficientes y que muchas veces los conductores no encuentran disponibilidad o no conocen los parqueaderos de la zona. Sin embargo cabe destacar que la administración manual de la información genera uno de los problemas más significativos que se pueden encontrar en los parqueaderos, como es la gran cantidad de tiempo que consume verificar la disponibilidad de los servicios cada vez que algún cliente los solicite, pues la verificación de los espacios logísticos a menudo tarda demasiado tiempo al ser buscados de forma personal. También en la elaboración de diversos recibos se suelen presentar demoras injustificadas y desgastes innecesarios de estos recursos, lo que le genera a la empresa pérdidas económicas y desprestigio en su imagen comercial.

Por otra parte, no hay que dejar de lado las filas extensas de carros a la entrada de sus espacios, el descontento de los conductores al darse cuenta de la inexistencia de un espacio disponible dentro del parque-automotor.

Muchas veces dudas en conducir a algún lugar por no saber si habrá un estacionamiento cerca, también se ha querido acceder a la información de la disponibilidad del parqueadero pero no había medios de cómo hacerlo. Gracias a estas circunstancias y demás problemas mencionados anteriormente, en lugar de un obstáculo, se convirtieron en la base de una nueva estrategia de negocios. Pensado en lo anterior, se Diseñó e implementó una apps web que muestra la disponibilidad en los parqueaderos, la cual apoya a todos los conductores y

personas a encontrar de manera fácil y rápida un lugar de estacionamiento disponible. La realización de este proyecto se llevó a cabo en la ciudad de Montería del departamento Córdoba.

Para desarrollar esta propuesta, se tomó como base o sustento, algunos trabajos realizados en otros países y aquí en Colombia, tales como: en la Universidad Católica de Santiago de Guayaquil – Ecuador, en la Facultad de Ingeniería, el estudiante de Ingeniería en Sistemas Computacionales Jonathan Grimaldo Tasigchana Villegas, desarrolló un Sistema Web online para informar la disponibilidad de parqueaderos públicos del sector bancario y comercial de la ciudad de Guayaquil denominador “Parqueo Seguro”. Este proyecto tiene como objetivo la creación de un sistema web que permitiera a los dueños de negocio destinados al aparcamiento de vehículos, darse a conocer a la ciudadanía en general a través del internet, así como también a la ciudadanía, el poder de acceder a información confiable sobre la ubicación (calles, direcciones, avenidas, entre otras), capacidad total de paqueos, disponibilidad de espacios libres, tarifas y horarios de atención mediante el uso de celulares o dispositivos que tengan acceso a internet. (Tasigchana Villegas, 2014)

Por otro lado en la Escuela Politécnica Nacional de Quito – Ecuador, los estudiantes Xavier Porras Torres y Javier Prado, desarrollaron un sistema prototipo de consulta de parqueaderos libres. El sistema ProPark es una herramienta de consulta que facilita a cualquier usuario que cuente con un PDA (Sistema operativo Windows Mobile), información referente a parqueaderos más cercanos a su ubicación y su disponibilidad. (Viracachá, 2012)

En la Universidad de los Andes – Colombia, los estudiantes Camilo Restrepo y Julián Espinel, desarrollaron una aplicación móvil, desarrollada para dispositivos Android, que permite mejorar los tiempos de búsqueda de un parqueadero, registro del vehículo y pago

de servicio, por parte del usuario al momento de hacer uso de un parqueadero público. Esta aplicación es denominada EasyPark, la cual pretende disminuir el tiempo de ingreso de un vehículo al parqueadero, el tiempo de pago por el tiempo de parqueo y una de las razones más importante es informar en tiempo real los cupos disponibles en cada parqueadero.

A Colombia llegó la aplicación Parking App, la aplicación para encontrar parqueadero. El lunes 21 de abril de 2014 llegó a Colombia la aplicación para dispositivos móviles que prometió atender las necesidades de los conductores en varias ciudades del país a la hora de encontrar parqueadero. Se trata de Parking App, está disponible en celulares con sistema operativo Android e iPhone, y los puede descargar en sus respectivas tiendas virtuales. Cuando abra la aplicación, si usted registra sus datos ésta le regalará 30 minutos gratis en cualquiera de sus parqueaderos. Su uso: si en el mapa satelital aparece un icono verde, quiere decir que la ocupación del parqueadero o valet parking es de 0 por ciento al 50 por ciento; si es de color gris es porque está cerrado. Si el icono es circular significa que es zona de parqueo, si es cuadrado es zona de valet parking; y finalmente el número que aparece dentro de cada figura es la tarifa del minuto que maneja el parqueadero. Entre los beneficios de la aplicación está encontrar rutas de acceso a través de los mapas de Google o Waze; informa en tiempo real la disponibilidad de los parqueaderos en la ciudad y la cercanía al conductor; horarios y hasta convenios. Además, puede filtrar información por ciudad, zonas, tarifas o tipo de establecimientos; también, puede marcar el vehículo para facilitar su ubicación y cronómetro.

La aplicación ofrece el servicio de valet parking con el que los usuarios podrán solicitar su vehículo directamente en línea y recibir notificaciones sobre el tiempo de espera de su entrega. Con Parking App puede acceder a promociones y descuentos para los

usuarios y pagos en línea, asimismo podrá encontrar esta aplicación a través [parking.net.co](http://parking.net.co) que cuenta con una oferta de 15 mil cupos de parqueo dotados con última tecnología. (cmi)

### 3. MARCO TEÓRICO

La tecnología y sus avances tienen como prioridad mejorar la calidad de vida de todos los seres humanos, utilizando herramientas esenciales las cuales causan un gran impacto en el modo de vivir de la sociedad haciendo de sus actividades diarias algo más fácil.

El entorno digital y el máximo crecimiento de las tecnologías han intervenido en unas variables de sectores a nivel mundial. Los avances en los sistemas de geolocalización toman cada vez con más poderío todos los sectores que benefician una sociedad, como plataforma que provee el desplazamiento a través de rutas a lugares muy transcurridos, ofreciendo una gama de servicios e información óptimos a los usuarios.

#### 3.1. SISTEMA DE INFORMACIÓN GEOGRÁFICA

Los SIG O GIS, en su acrónimo inglés geographic information system se han convertido en una herramienta indispensable al momento de gestionar recursos relacionados con la combinación de hardware, software y datos geográficos, ya que permiten capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográficamente con el objetivo de resolver problemas relacionados con los sistema de coordenadas terrestre. (Ibañez A. , 2008)

En la actualidad estos sistemas ofrecen un gran número de aplicaciones entre las que podemos resaltar:

**Localización de Elementos:** Los SIG permiten crear mapas virtuales sobre los que se pueden ubicar y localizar elementos, utilizando para ello georreferenciación y geolocalización.

**Cálculos de Rutas:** Los SIG se pueden utilizar para realizar una serie de cálculos que permiten conocer la ruta más óptima entre diferentes puntos o coordenadas del mapa.

**Cálculo de Distancias:** Los SIG se utilizan para cocer las distancias que hay entre diferentes puntos de un mapa delimitar un radio alrededor de una entidad.

**Mapas Cuantitativos:** Estos mapas son utilizados para delimitar ciertas zonas geográficas a través de su población y así localizar lugares que reúnen ciertos criterios demográficos y de esta manera tomar decisiones precisas.



**Ilustración 1:** Estructura de un SIG, Fuente: *laciudadvivaoirg*

Dentro del amplio aspecto de características pueden detectarse 3 puntos convergentes que afirman que el SIG:

- Es un sistema de información, compuesto por hardware, software, datos, procedimientos y recursos humanos, destinado a soportar los procesos de toma de decisiones.



- Trabaja con una base de datos espaciales alfanuméricos y cartográficos.
- Cuenta con funciones especializadas de captura, almacenamiento, transformación, modelización, análisis y presentación de datos espaciales, para la resolución de problemas de naturaleza geográfica.

### **3.1.1. GEORREFERENCIACIÓN**

Se refiere a las características que tiene un objeto al posicionarse sobre un mapa digital ya sea mediante un punto, vector, área, volumen, entre otras, sobre un mismo sistema de coordenadas geográfico. (Ibañez A. , 2008)

Al momento de realizar cualquier proceso de georreferenciación se deben tener en cuenta los siguientes pasos:

- ✓ Localizar y trazar el punto que es motivo de georreferenciación.
- ✓ Asignar un valor a este punto, de manera que quede identificado de forma univoca.
- ✓ Registrar los datos que acompañan al punto georreferenciado.

En este proyecto se usaran coordenadas geográficas, latitud y longitud, para georreferenciar todos los puntos de interés que acompañan a los parqueaderos. Este proceso de georreferenciación requiere un gran esfuerzo, sobre todo en el momento de recolectar la información, ya que se debe manejar un formato valido de las direcciones de los parqueaderos para que la ubicación sea lo más exacta posible en todos los puntos de interés que estén involucrados en el proyecto.

Como ya se ha comentado anteriormente, en este sistema se realizará la georreferenciación de objetos haciendo uso de la tecnología de Google Maps, la cual proporciona tanto los mapas topográficos, como el acceso a su índice geográfico. Google Maps no permite la

conexión directa a una base de datos y trabaja con archivos en formato JSON o XML para que la aplicación los analice o procese, por ende, se ha generado un formato de salida json con el resultado de las consultas que se realicen a la base de datos.

### **3.1.2. GEOCODIFICACIÓN**

Es el proceso de transformar una dirección interpretable por humanos en una ubicación de un mapa con el uso de coordenadas geográficas es decir latitud y longitud y viceversa. En el ámbito de los Sistemas de Información Geográfica se pueden encontrar con dos tipos de geocodificación las cuales son: geocodificación directa y geocodificación inversa.

#### **3.1.2.1. GEOCODIFICACIÓN INVERSA**

Es el proceso de encontrar una dirección a partir de coordenadas geográficas (latitud y longitud) este tipo de Geocodificación ubica de manera más exacta el punto de búsqueda que la geocodificación directa.

En este proyecto se utilizará ambas geocodificación, proporcionada por la API de Google Maps, para situar sobre el mapa los marcadores de los parqueaderos los cuales se generan a partir de la geocodificación directa, ya que para esto se toman las direcciones de los parqueaderos. La geocodificación inversa se utiliza para dibujar los marcadores del punto de origen de la emergencia, ya que esta utilizara los sistemas GPS de los móviles.

### **3.1.3. GEOLOCALIZACIÓN**

Cuando se habla de geolocalización se hace referencia específicamente al punto donde se encuentra posesionada una persona o cualquier tipo de objetos en el espacio geográfico, ya

sea por medio de coordenadas o un conjunto de puntos la posición actual y real en la que se encuentra una persona, lugar o cosa. (Chávez, 2011).

Este proceso es generalmente empleado por los sistemas de información geográfica, un conjunto organizado de software y hardware, más datos geográficos, que se encuentra diseñado especialmente para capturar, almacenar, manipular y analizar en todas sus posibles formas la información geográfica referenciada, con la clara misión de resolver problemas de gestión y planificación.

Entre ese amplio abanico de opciones que nos permiten descubrir la geolocalización se destacan los teléfonos móviles de alta gama, que son aquellos que cuentan con funciones especializadas y que por su portabilidad nos permiten conocer nuestra ubicación y actualizarla a medida que nos vamos movilizandoy por tanto cambiando de ubicación geográfica, estos en particular traen integrados receptores de GPS, los cuales y gracias a la red de satélites que rodea al planeta podrán ubicarnos en cualquier parte del globo terráqueo en el cual nos encontremos.

Otra alternativa ampliamente usada y a disposición de aquellos que no poseen celulares como los mencionados líneas arriba u otros equipos portátiles es la del Google Earth y Google Maps, que consiste en un programa informático, similar al Sistema de Información Geográfica, que permite obtener imágenes del planeta en tecnología 3D en combinación con imágenes de satélite, mapas y el motor de búsqueda de Google y así facilita la visualización de imágenes a escala del lugar del planeta tierra que se pretenda descubrir o identificar. (ABC, s.f.)

### **3.1.3.1. FUNCIONAMIENTO DE GEOCALIZACION EN MOVILES.**

La mayoría de los móviles inteligentes incorporan un chip GPS, que utiliza la información del satélite para calcular la posición exacta de la persona a través de servicios de Google Maps, el cual posiciona al usuario en un lugar. Cuando la señal GPS no está disponible, estas aplicaciones pueden utilizar la información de la antena de teléfono móvil para triangular y hacer una aproximación de donde está el usuario. Un sistema no tan perfecto como el GPS, pero que ha mejorado en los últimos años. Algunos sistemas de geocalización utilizan GPS y la triangulación celular (e incluso las redes WI-FI) de manera conjunta en lo que se denomina Assited GPS (A-GPS).

Cuando el cielo está despejado, la aplicación de geocalización en el móvil puede establecer la posición del usuario de manera precisa. Sin embargo, en interiores este posicionamiento falla más y en los locales donde las exhibiciones están muy próximas, es el usuario el que ha de seleccionar manualmente su posición exacta. Sin embargo se espera que en el futuro y con un mayor desarrollo de los sistemas A-GPS, se afine más en la posición geográfica del usuario cuando se encuentran en lugares cerrados como centros comerciales.

Los dispositivos móviles poseen varios subsistemas internos a estos, que posibilitan la ubicación en cualquier parte geográfica ya sea por su GPS y si no poseen por su identificador celular (GSM, G3S) o quizás por la conectividad de internet.

## **3.2. GOOGLE MAPS**

Google Maps es un servicio ofrecido por Google que consiste en un servidor de aplicaciones de mapas en la web que ofrece la posibilidad de observar mapas y fotos satelitales de todo el planeta, así como puntos específicos como hospitales o almacenes

importantes a través de una página web. Este servicio es totalmente gratuito y permite a los usuarios integrar muchas de sus características a su sitio Web. (Ibañez, 2013).

Desde su lanzamiento en febrero de 2005, la aplicación cartográfica de Google ha conmocionado a la comunidad de desarrolladores. Si bien sus principios técnicos de base eran ya conocidos, incluso utilizados desde hacía tiempo, la aplicación de Google los combina de manera inteligente, y sobre todo ofrece una accesibilidad sin igual.

Varios aspectos de Google Maps es la facilidad de uso por cualquier usuario: el sistema de deslizamiento de imagen, acoplado a la carga dinámica de nuevas imágenes; la adaptación del mapa al tamaño de ventana del navegador; la interfaz minimalista; la posibilidad de cambiar de tipo de mapa en un clic.

Como todas las demás aplicaciones de Google, Google Maps descansa poderosamente sobre la utilización de JavaScript. La carga y el deslizamiento de imagen no podrían efectuarse sin este código. Según el nivel de zoom, un mapa podría descomponerse en varias decenas de miles de cuadrados. Esta descomposición se realiza automáticamente por un script del lado servidor. Cuando el utilizador hace deslizar la imagen en un sentido, JavaScript calcula cuales son los cuadrados involucrados, envía al servidor una pregunta sobre las imágenes en cuestión, y las coloca alineadas junto a las otras.

Cada cuadrado es almacenado en un fichero cuyo nombre indica su longitud, su latitud, y el valor del zoom. Recuperar estas informaciones para todos los cuadrados a colocar, no es sino una cuestión de derivación de los datos conocidos para un solo cuadrado.

Todo ello necesita, por tanto, una sincronización perfecta entre lo que puede preguntar el código JavaScript, y lo que está efectivamente disponible en el servidor. El programa de recorte inicial de la imagen completa es por tanto extremadamente importante, porque es el responsable, para la precisión del corte y el rigor de su etiquetamiento, del buen desenvolvimiento de los acontecimientos desencadenados por las acciones del usuario.

Google Maps provee a los desarrolladores un API capaz de aprovechar los datos disponibles a través del servicio, en el seno de las propias aplicaciones. Y más interesante para el funcionamiento mismo del servicio, *Google* da acceso a AjaXSLT, un proyecto Open Source que propone una biblioteca JavaScript.

Con las API de Google Maps publicadas, numerosos sitios han aparecido proponiendo sus propias maneras de extraer los datos de los mapas de Google y presentarlos en la página a través de scripts específicos, proponiendo un servicio particular.

### 3.2.1. TIPOS DE MAPAS

Para expresar el tipo de mapa inicial hacemos lo siguiente:

```
mapTypeId: google.maps.MapTypeId.ROADMAP
```

Se admite los siguientes tipos de mapas:

- **Roadmap:** que muestra los mosaicos normales en 2D predeterminados de Google Maps.
- **Satellite:** muestra imágenes de satélites.

- **Hybrid:** Muestra una mezcla de mosaicos fotográficos y una capa de mosaico para los elementos del mapa más destacados (carreteras, nombres de ciudades, etc.).
- **Terrain:** Muestra mosaico de relieve físico para indicar las elevaciones del terreno y las fuentes de agua (montañas, ríos, etc.).

### 3.2.2. API DE GOOGLE MAPS

Una forma de visualización de las posiciones o puntos de coordenadas geográficas, que usualmente se utilizan es el servicio que ofrece Google. Google Maps nos ofrece imágenes vía satélite de todo el planeta, combinadas en el caso de algunos países, con mapas de sus ciudades, lo que ha aumentado las utilidades ofrecidas desde numerosas páginas web.

Como todas las demás aplicaciones Google, Google Maps descansa poderosamente sobre la utilización de JavaScript, la carga y el deslizamiento de imagen no podrían efectuarse sin este código. El API de Google Maps proporciona estos servicios web como una interfaz para solicitar a servicios externos datos del API de Google Maps y utilizarlos en las aplicaciones de google Maps. Estos servicios se han diseñado para utilizarse en combinación con un mapa. Los datos enviados a Google Maps para la visualización son obtenidos por medio del software de los dispositivos GPS y este genera un archivo plano de la posición del individuo y otros ítems relevantes para el caso. Posteriormente estos datos serán mostrados a través de la API de Google en el aplicativo final con sus coordenadas geográficas y rutas. (Ibañez, 2013)

### **3.2.2.1. SERVICIOS WEB DEL API DE GOOGLE MAPS**

El API de Google Maps proporciona estos servicios web como una interfaz para solicitar a servicios externos datos del API de Google Maps y utilizarlos en las aplicaciones de google Maps. Estos servicios se han diseñado para utilizarse en combinación con un mapa.

Estos servicios web envían solicitudes HTTP a URL específicas, transmitiendo a los servicios parámetros de URL como argumentos. Normalmente, estos servicios devuelven los datos de la solicitud HTTP en formato JSON o XML para que tu aplicación los analice o procese.

### **3.2.3. API DE MATRIZ DE DISTANCIA**

Es un servicio que ofrece Google Maps que permite calcular la distancia y el tiempo para una matriz de orígenes y destinos. La información devuelta se compone de dos filas que incluyen los valores de duración y de distancia para cada par de puntos, esta Api permite obtener una distancia mucho más exacta entre el origen y el destino, puesto que tiene en cuenta que estos cálculos se realicen sobre las autopista, y no como muchos algoritmos que calculan las distancias sin tener en cuenta que se encuentre un río o una montaña en medio de estas. Otras de las ventajas que tiene usar este servicio es que permite configurar los cálculos a partir del transporte que se vaya a utilizar para trasladarte entre el punto origen y el punto destino, dando como resultado final la distancia y el tiempo que durara el recorrido entre puntos. (Developers, 2013)

#### **3.2.3.1. LIMITES DE USO**



Cada consulta enviada al API de matriz de distancia está limitada por el número de elementos permitidos, que se determina multiplicando el número de orígenes por el número de destinos.

El API de matriz de distancia cuenta con los siguientes límites:

- 100 elementos por consulta,
- 100 elementos cada 10 segundos,
- 2.500 elementos en un período de 24 horas.

### **3.3.APPS WEB**

Es una versión de la página web, la cual es optimizada gracias a HTML5 y CSS3, y es adaptable a cualquier dispositivo móvil. Es una página que se puede abrir desde el navegador de cualquier terminal independiente del sistema operativo que utilice.

Su principal ventaja es que tiene la capacidad de adaptarse a cualquier dispositivo móvil, solo es suficiente con que el dispositivo cuente con un navegador móvil actualizado. Sin embargo para visualizar la página de forma correcta en las diferentes plataformas, es necesario incluir algún fragmento de códigos especiales para cada una.

La Apps Web no ocupa memoria en los dispositivos ya que no es una App que se pueda descargar desde los Market Places, no le hacen falta actualizaciones ya que siempre se accederá a la última versión de la página como cualquier página web, al no ser una aplicación nativa, el consumo de recursos en los dispositivos es mínima. (QODE, 2014)

Las interfaces web tienen ciertas limitaciones en las funcionalidades que se ofrecen al usuario. Hay funcionalidades comunes en las aplicaciones de escritorio como dibujar en la pantalla o arrastrar-y-soltar que no están soportadas por las tecnologías web estándar. Los desarrolladores web generalmente utilizan lenguajes interpretados (scripts) en el lado del

cliente para añadir más funcionalidades, especialmente para ofrecer una experiencia interactiva que no requiera recargar la página cada vez. Recientemente se han desarrollado tecnologías para coordinar estos lenguajes con las tecnologías en el lado del servidor. Como ejemplo, AJAX es una técnica de desarrollo web que usa una combinación de varias tecnologías.

Existen numerosos lenguajes de programación empleados para el desarrollo de aplicaciones web en el servidor, entre los que destacan: PHP, Java con sus tecnologías Java Servlets y JavaServer Pages (JSP), Javascript, Perl, Ruby, Python; C# y Visual Basic con sus tecnologías ASP/ASP.NET. También son muy utilizados otros lenguajes o arquitecturas que no son propiamente lenguajes de programación, como HTML o XML.

Una aplicación web está normalmente estructurada como una aplicación de tres-capas. En su forma más común, el navegador web ofrece la primera capa, y un motor capaz de usar alguna tecnología web dinámica, por ejemplo: PHP, Java Servlets o ASP, ASP.NET, CGI, ColdFusion, embPerl, Python o Ruby on Rails que constituye la capa intermedia. Por último, una base de datos constituye la tercera y última capa.

El navegador web manda peticiones a la capa intermedia que ofrece servicios valiéndose de consultas y actualizaciones a la base de datos y a su vez proporciona una interfaz de usuario.



**Ilustración 2:** Representativa de Apps Web, Fuente: <http://rho-design.com/project/creacion-de-apps-y-web-apps/>

### 3.4. BASES DE DATOS

Son bancos de información que contienen datos relativos a diversas temáticas y categorizados de distinta manera, pero que comparten entre sí algún tipo de vínculo o relación que busca ordenarlos y clasificarlos en conjunto.

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido; una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital, siendo este un componente electrónico, y por ende

se ha desarrollado y se ofrece un amplio rango de soluciones al problema del almacenamiento de datos.

Existen programas denominados sistemas gestores de bases de datos, abreviado DBMS, que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos DBMS, así como su utilización y administración, se estudian dentro del ámbito de la informática.

#### **3.4.1. GESTOR DE BASE DE DATOS**

Un gestor de base de datos o sistema de gestión de base de datos (SGBD o DBMS) es un software que permite introducir, organizar y recuperar la información de las bases de datos, es decir, administrarlas. El propósito general de los sistemas de gestión de bases de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización (Martínez Orozco, 2013). Entre estos sistemas gestores de bases de datos tenemos: PostgreSQL, MySQL, SQL Server, Oracle, entre otros.

Estos sistemas también proporcionan métodos para mantener la integridad de los datos, para administrar el acceso de usuarios a los datos y para recuperar la información si el sistema se corrompe. Permiten presentar la información de la base de datos en variados formatos. La mayoría incluyen un generador de informes. También pueden incluir un módulo gráfico que permita presentar la información con gráficos y tablas.

Generalmente se accede a los datos mediante lenguajes de interrogación, lenguajes de alto nivel que simplifican la tarea de construir las aplicaciones. También simplifican la interrogación y la presentación de la información. Un SGBD permite controlar el acceso a los datos, asegurar su integridad, gestionar el acceso concurrente a ellos, recuperar los datos

tras un fallo del sistema y hacer copias de seguridad. Las bases de datos y los sistemas para su gestión son esenciales para cualquier área de negocio, y deben ser gestionados con esmero.

### **3.4.2. MySQL**

Es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. Es muy utilizado en aplicaciones web, como Drupal o phpBB, en plataformas (Linux/Windows – Apache – MySQL – PHP/Perl/Python), y por herramientas de seguimiento de errores como Dugzilla.

“MySQL es un sistema gestor de bases de datos relacionales rápido, sólido y flexible. Es idóneo para la creación de bases de datos con acceso desde páginas web dinámicas, así como para la creación de cualquier otra solución que implique el almacenamiento de datos, posibilitando realizar múltiples y rápidas consultas. Está desarrollado en C y C++, facilitando su integración en otras aplicaciones desarrolladas también en esos lenguajes. Es un sistema cliente/servidor, por lo que permite trabajar como servidor multiusuario y de subprocesamiento múltiple, o sea, cada vez que se crea una conexión con el servidor, el programa servidor establece un proceso para manejar la solicitud del cliente, controlando así el acceso simultáneo de un gran número de usuarios a los datos y asegurando el acceso a usuarios autorizados solamente. Es uno de los sistemas gestores de bases de datos más utilizado en la actualidad, utilizado por grandes corporaciones como Yahoo! Finance, Google, Motorola, entre otras”. (EcuRed, 2014).

FORTALEZAS DE MySQL	
• Tiene gran penetración en el mercado en los servidores de datos orientados a WEBSITES.	
• Se distribuye bajo licencia GNU/GPL, lo que lo hace en ciertas circunstancias gratuito; por ello es adoptado por millones de usuarios para sus páginas personales.	
• Fue diseñado para acceso rápido a información, por lo cual usa un sistema de archivo de almacenamiento plano, que aumenta la eficiencia de la lectura.	
• Excelente gestor de bases de datos, dando un sistema de estructuración de información (tablas, columnas, índices, etc) y una interfase SQL para poder acceder a ello.	
• Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.	
• Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferentes velocidades de operación, soporte físico, capacidad, distribución geográfica, transacciones, entre otras.	
• Conectividad segura	
• Búsqueda e indexación de campos de texto.	
• Transacciones y claves foráneas.	
• Ante la ventaja de tener una velocidad de respuesta rápida, incorpora elementos que le faltaban: <ul style="list-style-type: none"> <li>○ InnoDB - para añadir integridad referencial y transaccionalidad al motor de BS</li> <li>○ Librerías API - para usarlas como paquetes embebidos en las aplicaciones</li> </ul>	

**Tabla 1:** Fortalezas de MySQL

### 3.5. SENSORES

Los sensores son dispositivos capaces de detectar magnitudes físicas o químicas que llamamos variables de instrumentación y transformarlas en variables o señales eléctricas. También se utilizan para detectar obstáculos mediante la emisión de radiación infrarroja,

estos son conocidos como los sensores de proximidad de propósito general, también los sensores infrarrojos pueden medir el calor de un objeto, así como para detectar el movimiento. Muchos de estos tipos de sensores miden sólo la radiación infrarroja, en lugar de emitirla, y por lo tanto se conocen como infrarrojo pasivo. Los sensores inductivos son de gran utilización hoy en día, ya sea por la industria en muchos de sus procesos para la detección de materiales ferrosos o en aplicaciones cotidianas para la sociedad, como en el control vehicular, como es nuestro caso de estudio.

Existen sensores que se clasifican según el tipo de señal de salida, el cual puede ser analógico, digital; y según la magnitud física a detectar, la cual es la posición, velocidad, fuerza y par, presión, caudal, proximidad, entre otros. En la siguiente tabla podemos ver los diferentes tipos de sensores:

Magnitud Física	Transductor	Características
Posición (Lineal o Angular)	Potenciómetro	Analógico
	Encoder	Digital
	Síncro y Resolver	Analógico
Pequeños desplazamientos	Transformador diferencial	Analógico
	Galga Extensiométrica	Analógico
Velocidad (Lineal o Angular)	Dinamo tacométrica	Analógico
	Encoder	Digital
	Detector inductivo u óptico	Digital
Aceleración	Acelerómetro	Analógico
	Sensor de velocidad + calculador	Digital
Fuerza y Par	Galga Extensiométrica	Analógico
Temperatura	Termopar	Analógico
	Resistencias PT100	Analógico
	Termistores CTN	Analógico
	Termistores CTP	Todo-Nada
	Bimetales	Todo-Nada
Sensores de Presencia o Proximidad	Inductivos	Analógico o Todo-Nada
	Capacitivos	Todo-Nada
	Ópticos	Analógico o Todo-Nada

**Tabla 2:** Tipos de Sensores (Balcells J., s.f.)

### **3.5.1. SENSORES FOTOELÉCTRICOS**

Los sensores fotoeléctricos son dispositivos electrónicos que usan un haz de luz para detectar la presencia o ausencia de un objeto. Estos sensores permiten la detección precisa y sin contacto de los objetos. Emite luz infrarroja, roja o laser, el objeto debe interrumpir el haz de luz o reflejarlo hacia el sensor para activar la salida del mismo. (Rodriguez, 2012)

El sistema está compuesto de dos componentes separados: un emisor y un receptor. El emisor irradia luz sincronizadas en infrarrojos o en área roja de longitud de onda. El receptor reconoce la llegada de la luz y produce inmediatamente una señal de interrupción provocada por un objeto en la zona de detección.

Algunas de las aplicaciones más comunes en las que se pueden utilizar sensores fotoeléctricos son:

- Detectar el paso de un producto.
- Controlar que la altura o la anchura de un producto sea la adecuada.
- En envasadoras, para detectar que una bobina de plástico o papel está agotada.
- Protección de zonas de riesgo, detectando intrusiones (puertas de acceso para personas o vehículos).
- Ajustar la distancia de un elemento móvil.
- Como final de carrera sin contacto.
- Ajuste de marcas de impresión Detectores de humo.

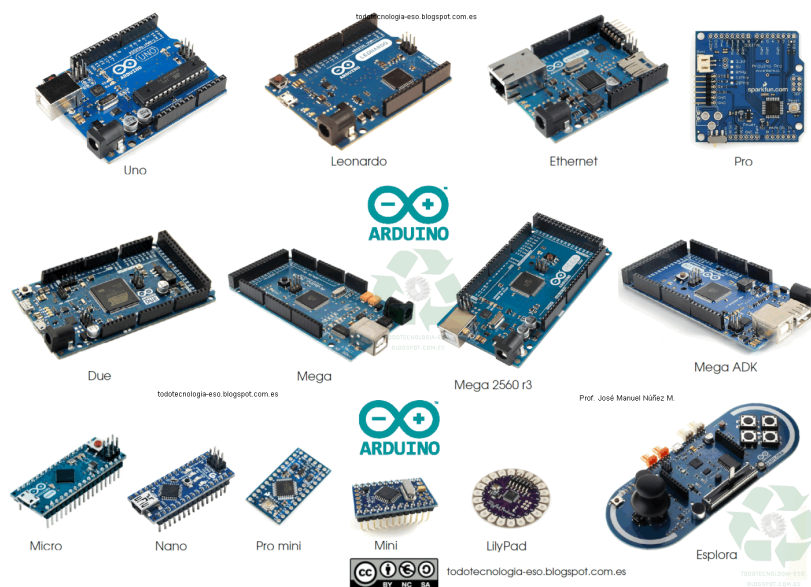




**Ilustración 3:** Sensores Fotoeléctricos, Fuente: [http://mco-s2-p.mlstatic.com/barrera-fotoelectronica-perimetral-doble-lente-5870-MCO5007799911\\_092013-O.jpg](http://mco-s2-p.mlstatic.com/barrera-fotoelectronica-perimetral-doble-lente-5870-MCO5007799911_092013-O.jpg)

### 3.6. ARDUINO

Es una placa de código libre y de hardware libre, es una placa que se puede comprar e incluso fabricarla nosotros mismos. Tiene una comodidad muy extensa con miles y miles de códigos libres accesibles y documentados para que podamos realizar todo tipo de proyecto, es realmente sencilla de utilizar la cual podemos conectar directamente por el puerto USB a nuestro ordenador, funciona con Windows y Linux. (cscjprofes.com, 2014)



**Ilustración 4:** Tipos de Arduinos. (Google, s.f.), Fuente: <http://todotecnologia-eso.blogspot.com.co/2014/12/que-modelo-arduino-comprar.html>

Esta plataforma electrónica abierta nos ayuda en la creación de prototipos basados en software y hardware flexible y factible de usar, se creó para artistas, diseñadores, aficionados y cualquier interesado en crear entornos u objetos interactivos.

Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea controlando luces, motores y otros actuadores. El micro controlador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno a desarrollo Arduino (basado en Processing). Los proyectos hechos con Arduinos pueden ejecutarse sin necesidad de conectar a un ordenador, si bien tiene la posibilidad de hacerlo y comunicar con diferentes tipos de software (Flash, Processing, MaxMSP).

### **3.6.1. ARDUINO UNO**

Es una placa electrónica, la cual cuenta con 14 pines digitales de entrada/salida, los cuales 6 de esos pines se pueden utilizar como salidas PWM, otros 6 de entradas analógicas, un cristal de cuarzo de 16 MHz, una conexión USB, un conector de alimentación, una cabecera ICSP y un botón de reinicio.

Es una plataforma computacional física de Open-Source, basada en una simple tarjeta de I/O y un entorno de desarrollo que implementa el lenguaje Processing/Wiring. El Arduino Uno se puede utilizar para desarrollar objetos interactivos o puede ser conectado a software de computadora. El IDE open-source puede ser descargado de forma gratuita y está disponible para Mac OS X, para Windows y Linux. (Cart, 2015).

### **3.6.2. ARDUINO SHIELD ETHERNET**

El Shield conecta un controlador Wiznet Ethernet a una placa Arduino vía el bus SPI (Serial Peripheral Interface) permitiendo conectarse a internet. El Wiznet ofrece una red (IP) apilar de TCP y UDP. Soporta hasta cuatro pines de conexión de socket simultáneas. Utiliza una biblioteca de Ethernet para escribir bocetos que se conectan a internet a través del escudo, este se conecta a una placa Arduino usando largas cabeceras wire-wrap que se extienden a través del escudo. (Pujol Perez, 2013)

## **4. MARCO CONCEPTUAL**

### **4.1. GPS**

El GPS (Global Positioning System: sistema de posicionamiento global) es un sistema global de navegación por satélite (GNSS) que permite determinar en todo el mundo la posición de un objeto, una persona o un vehículo con una precisión hasta de centímetros (si se utiliza GPS diferencial), aunque lo habitual son unos pocos metros de precisión. El sistema fue desarrollado, instalado y actualmente operado por el Departamento de Defensa de los Estados Unidos. El sistema GPS funciona mediante una red de 24 satélites en órbita sobre el globo, a 20.200 kph de altura, con trayectorias sincronizadas para cubrir toda la superficie de la Tierra. (Wikipedia.org, 2015)

### **4.2. HTML**

HTML es un lenguaje de marcas de texto o de etiquetas utilizado para crear las páginas web. Con este lenguaje se desarrolla una representación sobre la estructura de los contenidos al igual que la de sus complementos como lo son imágenes, animaciones, entre otros. (Alvarez, 2001)

### **4.3. CSS3**

CSS son las siglas de Cascading Style Sheets, en español Hojas de estilo en Cascada. CSS, es un lenguaje de estilo que define la presentación de los documentos HTML permitiéndole a los usuarios personalizar las páginas web implementando un sin número de propiedades para el diseño y por lo tanto darle las formas, tamaños y color a las páginas web de una manera más armónica. (html.net, 2013).

#### **4.4. LENGUAJE DE PROGRAMACIÓN PHP**

(Acrónimo de PHP: Hypertext Preprocessor): Lenguaje de programación enfocado a la creación de páginas web dinámicas. Es ejecutado principalmente del lado del servidor. PHP es un lenguaje ampliamente usado y que está diseñado especialmente para desarrollo web. (maestrosdelweb.com, 2001)

#### **4.5. JAVA**

Es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo, lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra.

#### **4.6. JAVASCRIPT**

Es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Se utiliza en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas<sup>4</sup> aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS).

#### **4.7. INTERFAZ**

Es lo que conocemos en inglés como *interface* (“superficie de contacto”). En informática se utiliza para nombrar a la conexión física y funcional entre dos sistemas o dispositivos de cualquier tipo dando una comunicación entre distintos niveles.

La palabra interfaz es utilizada en distintos contextos como: Interfaz como instrumento, Interfaz como superficie e Interfaz como espacio.

#### **4.8. C++**

Es un lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup. La intención de su creación fue el extender al lenguaje de programación C mecanismos que permiten la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido.

#### **4.9. ICSP**

Es el acrónimo de la frase en inglés:”In Circuit Serial Programming” (Programación Serial En Circuito); es una tecnología incluida en todos los microcontroladores PIC de Microchip más recientes y posibilita la reprogramación de los mismos sin que sea necesaria la remoción de éstos de su circuito de aplicación.

#### **4.10. PWM**

Modulación por ancho de pulsos (siglas en inglés de pulse-width modulation) de una señal o fuente de energía es una técnica en la que se modifica el ciclo de trabajo de una señal periódica.

#### **4.11. LENGUAJE PROCESSING**

Lenguaje de programación orientado a la creación visual y plataforma de creación de aplicaciones computacionales altamente interactivas que no solo se alimentan de los eventos producidos por los usuarios, si no por una gran cantidad de librerías que permiten conectarse a un sin número de plataformas y aplicativos que enriquecen las aplicaciones y le permiten producir una gran cantidad de acciones sobre el mundo virtual de la computadora.

#### **4.12. WIRING**

Es un entorno de programación de entradas/salidas de código abierto para explorar las artes electrónicas, los medios materiales, la enseñanza y el aprendizaje de la programación informática y creación de prototipos con electrónica.

## **5. METODOLOGIA**

Para llevar a cabo el proceso de diseño e implementación de la Apps web que muestre la disponibilidad en los parqueaderos de la ciudad de Montería del departamento de Córdoba, se puso en marcha una metodología de trabajo en la cual se describió el tipo de investigación a la que pertenece nuestro proyecto, las diferentes técnicas y procedimientos utilizados para cumplir con el objetivo general del proyecto.

La metodología es de forma descriptiva y tecnológica aplicada; a nivel descriptivo se estudiaron las técnicas que permitieron adquirir la información necesaria para el desarrollo del proyecto, donde se pudieron observar propiedades importantes como lo es la disponibilidad de los parqueaderos. Técnicamente el desarrollo de la investigación tecnológica aplicada corresponde al diseño, construcción y pruebas del sistema como producto final.

La población objeto a la cual se dirige el proyecto son los parqueaderos ubicados en el centro de la ciudad de Montería del departamento de Córdoba, a los cuales está dirigido u orientado nuestro proyecto, puesto que estos serán los beneficiarios.

La línea de investigación en la que está enfocado el proyecto, es la ingeniería de software, ya que permitió implementar la construcción de un software optimo, eficiente e intuitivo, el cual permitió abarcar todas las necesidades y así dar garantías suficientes para la viabilidad y calidad del software.



## 5.1. FASES DEL PROYECTO

El proceso de investigación se realizara en las siguientes fases:

FASES	OBJETIVO DE LA FASE	Descripción
FASE I	Análisis del área de aplicación del proyecto	Para la realización del proyecto fue necesario indagar acerca de la verdadera problemática que afrontan miles de conductores en todo el territorio regional a la hora de parquear y la mejor manera de contribuir a una solución eficaz e innovadora que se adaptara a las condiciones y necesidades tanto de los usuarios (conductores) como de los parqueaderos.
FASE II	Estudio de factores que hacen parte de la problemática	El estudio de los factores que se ven involucrados en la problemática se realizó de manera racional teniendo en cuenta el contexto regional y geográfico, esto con el fin de que la problemática fuera resuelta de una manera sencilla, pero que conllevara a un impacto real y tangible.
		Los requisitos para la realización del proyecto se hicieron determinando el

<b>FASE III</b>	<b>Análisis y especificación de requisitos</b>	propósito u objetivo del proyecto el cual es diseñar e implementar una Apps Web que muestre la disponibilidad en los parqueaderos utilizando la api de Google Maps que permite mejorar el tiempo de respuesta entre usuarios y los parqueaderos.
<b>FASE IV</b>	<b>Diseño del sistema que se aplicará para la solución de la problemática</b>	El diseño del sistema se realizó teniendo en cuenta la necesidad y la accesibilidad a la tecnología que tienen los usuarios ya que se desarrolló una Apps Web que podrá ser usada por los usuarios para conocer la disponibilidad en los parqueaderos. Por otro lado los administradores de cada establecimiento (parqueadero) utilizaran la aplicación para tener control y acceso a la información de cada uno de los autos o clientes que requieren el servicio.

**Tabla 3:** Fases de investigación

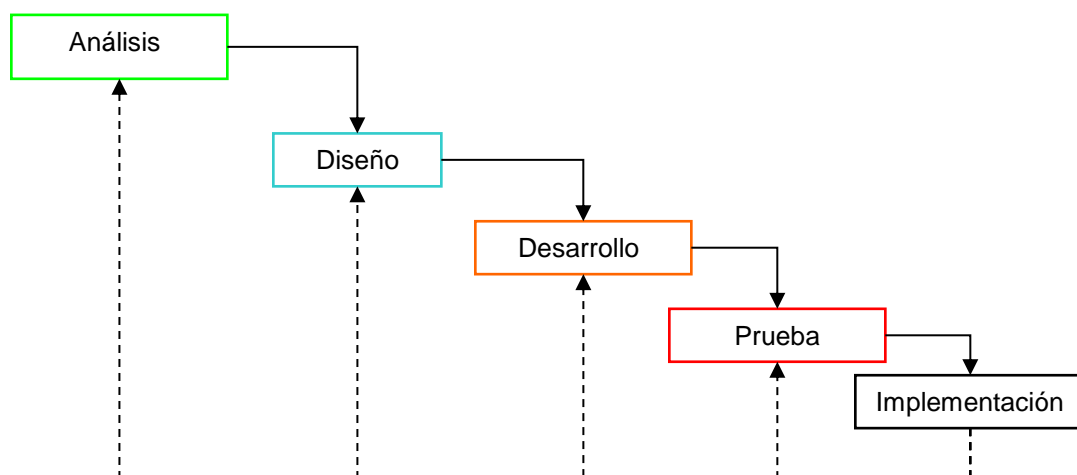
## 5.2. METODOLOGÍA DEL DESARROLLO DEL PRODUCTO

Para la realización y puesta en marcha de Easy Parking nos basamos en el Modelo en cascada. El cual es denominado en “cascada” debido a la posición de las fases en el desarrollo de esta, que parecen caer en cascada.

El Modelo de desarrollo en cascada es el enfoque metodológico que ordena rigurosamente las etapas del proceso para el desarrollo de software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior.

Al final de cada etapa, el modelo está diseñado para llevar a cabo una revisión final, que se encarga de determinar si el proyecto está listo para avanzar a la siguiente fase.

Siguiendo los parámetros del modelo en cascada nuestro equipo dividió el número de actividades y objetivos de acuerdo al cronograma de actividades, las mejoras y adiciones a la aplicación se hacían semana tras semana para luego ser mostradas al asesor para que este diera la aprobación de mejoras cerrando así cada ciclo del modelo.



**Ilustración 5:** Modelo Cascada, Fuente: Diseño propio.

ETAPAS	DESCRIPCIÓN
<b>ETAPA 1: ANÁLISIS</b>	<p>Creación de un modelo entidad relación. Seguido de esto se realizaron los casos de uso respectivos, definiendo roles de los actores del sistema. Así como también se definieron las clases con sus respectivos atributos funciones y tipos de datos.</p>
<b>ETAPA 2: DISEÑO</b>	<p>Utilizando Bootstrap se realizó la maquetación de las primeras vistas de Easy Parcking (Apps Web), luego utilizando el lenguaje de programación PHP se realizó la codificación y las respectivas conexiones a la base de datos.</p> <p>Para la realización de la Apps web fue necesario la utilización de: HTML5, CSS3 y Bootstrap (Maquetación y Diseño), PHP y Javascript (Consultas con el servidor y validación de formularios).</p> <p>Además fue necesario el uso del API de Google Maps para la geolocalización de usuarios y parqueaderos del sistema.</p>
<b>ETAPA 3: DESARROLLO</b>	<p>Utilizando Bootstrap se realizó la maquetación de las primeras vistas de Easy Parcking (Apps Web), luego utilizando el lenguaje de programación PHP se realizó la codificación y las respectivas conexiones a la base de datos.</p> <p>Para la realización de la plataforma web fue necesario la utilización de: HTML5, CSS3 y Bootstrap (Maquetación y Diseño), PHP y Javascript (Consultas con el servidor y validación de formularios).</p>

	Además fue necesario el uso del API de Google Maps para la geolocalización de usuarios y parqueaderos del sistema.
<b>ETAPA 4: PRUEBA</b>	Se constató que todas las funciones y formularios cumplan con los requerimientos del sistema, validando que los datos ingresados y procesados por la Apps web estén correctos y sean acordes al establecido previamente en los diagramas de secuencia y casos de uso.

**Tabla 4:** Etapas de desarrollo del producto

## 6. DESARROLLO

En esta sección presentaremos diversas ilustraciones, las cuales explicaran la forma como está constituido el sistema, desde el punto de vista conceptual y de diseño, así como también la manera en que se da el tratamiento de los datos y el conjunto de acciones que se generan mediante la interacción con los actores del sistema.

### 6.1. ARQUITECTURA DEL SISTEMA

El sistema desarrollado muestra la cantidad de puestos disponibles que hay en los parqueaderos del centro de la ciudad de Montería, este sistema está compuesto por una Apps Web que podrá ser accedida por cualquier tipo de navegador que utilice el usuario que desea obtener la información.

El sistema se implementó con la arquitectura de tres capas que se encuentran representadas por el patrón MVC (Modelo Vista Controlador), permitiendo el acceso a la información a Usuarios, a administradores de la App, administradores del parqueadero y porteros de los mismos.

El MVC se encarga de definir en qué bloque o capa se estructurará lógicamente la aplicación, detallando de esta forma las responsabilidades exactas de cada capa y la forma que tienen para relacionarse entre sí. (Bahit, 2013)

A continuación se especificaran las capas del modelo.

- **Capa de Persistencia:** En esta capa se encuentra el código que permite acceder a la fuente de datos, ella trabaja esencialmente con cuatro operaciones básicas (insertar, recuperar, actualizar y eliminar), que se realizan sobre una base de

datos relacional utilizando el sistema de gestión o motor de base datos MySQL. (Bahit, 2013).

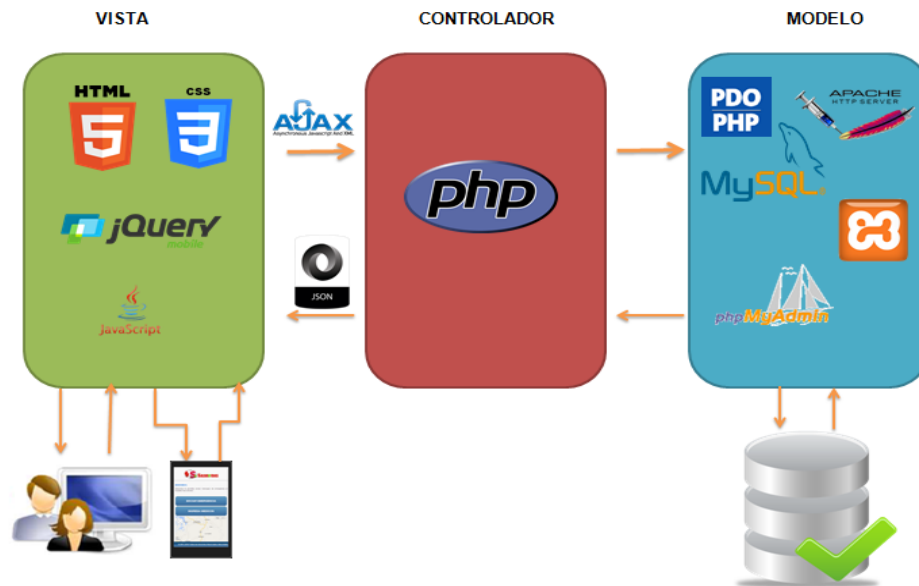
- **Capa de Lógica del Negocio:** En esta capa se encuentra el código encargado de recibir las peticiones del usuario, en ella se realiza la instancia de cada una de las clases que se encuentran en la capa de persistencia y se realizan las validaciones necesarias para recibir y enviar datos entre la capa de persistencia y la capa de presentación. (Bahit, 2013)
- **Capa de Presentación:** En esta capa se encuentra las interfaces graficas (formularios, tablas, cajas de textos, etc.) que le permitirán al usuario interactuara con el sistema, además cuenta con validaciones incorporadas por html5 y JavaScript. (Bahit, 2013)

Como ya se ha mencionado la implementación de una arquitectura de tres capas está relacionada de manera directa con el patrón MVC el cual se encarga de separar la lógica de negocio de la interfaz del usuario y es el más utilizado en aplicaciones Web, ya que facilita la funcionalidad, el mantenimiento y escalabilidad del sistema, de forma simple y sencilla.

El Modelo Vista Controlador o MVC divide las aplicaciones en tres niveles de abstracción:

- **Modelo:** Es el encargado de interactuar con la base datos gestionando todos los accesos a dicha información, tanto consultas como actualizaciones.
- **Vista:** La vista es la encargada de mostrar los datos al usuario de manera agradable al ojo humano.

- **Controlador:** Es el intermediario entre la vista y el modelo. Es quien controla las interacciones del usuario solicitando los datos al modelo y entregándolos a la vista para que ésta, lo presente al usuario.



**Ilustración 6:** Arquitectura del sistema MVC y las herramientas utilizadas en cada una de las capas del modelo.

### 6.1.1. ARQUITECTURA DEL DISPOSITIVO

El dispositivo que nos ayuda a obtener la información de la disponibilidad de los parqueaderos está conformado con diversos componentes electrónicos tales como:

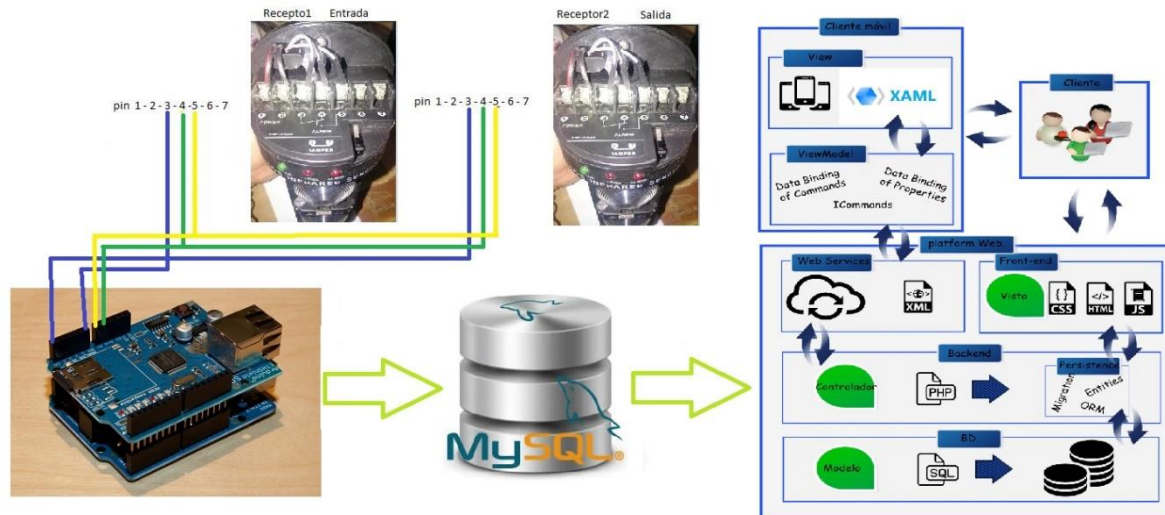
- Barreras Fotoeléctricas (Receptor y Emisor)
- Arduino Uno
- Shield Ethernet
- Bases de datos



Las barreras fotoeléctricas (Emisor y Receptor) traen 7 pines. En el 1 y 2 es la conexión VCC o alimentación, los pines 3, 4 y 5 son los que proporcionan la señal de alarma, y en nuestro caso son los pines que nos dan la señal de entrada o salida de un vehículo, estos pines van conectados a la Shield Ethernet que a su vez está conectada con el Arduino.

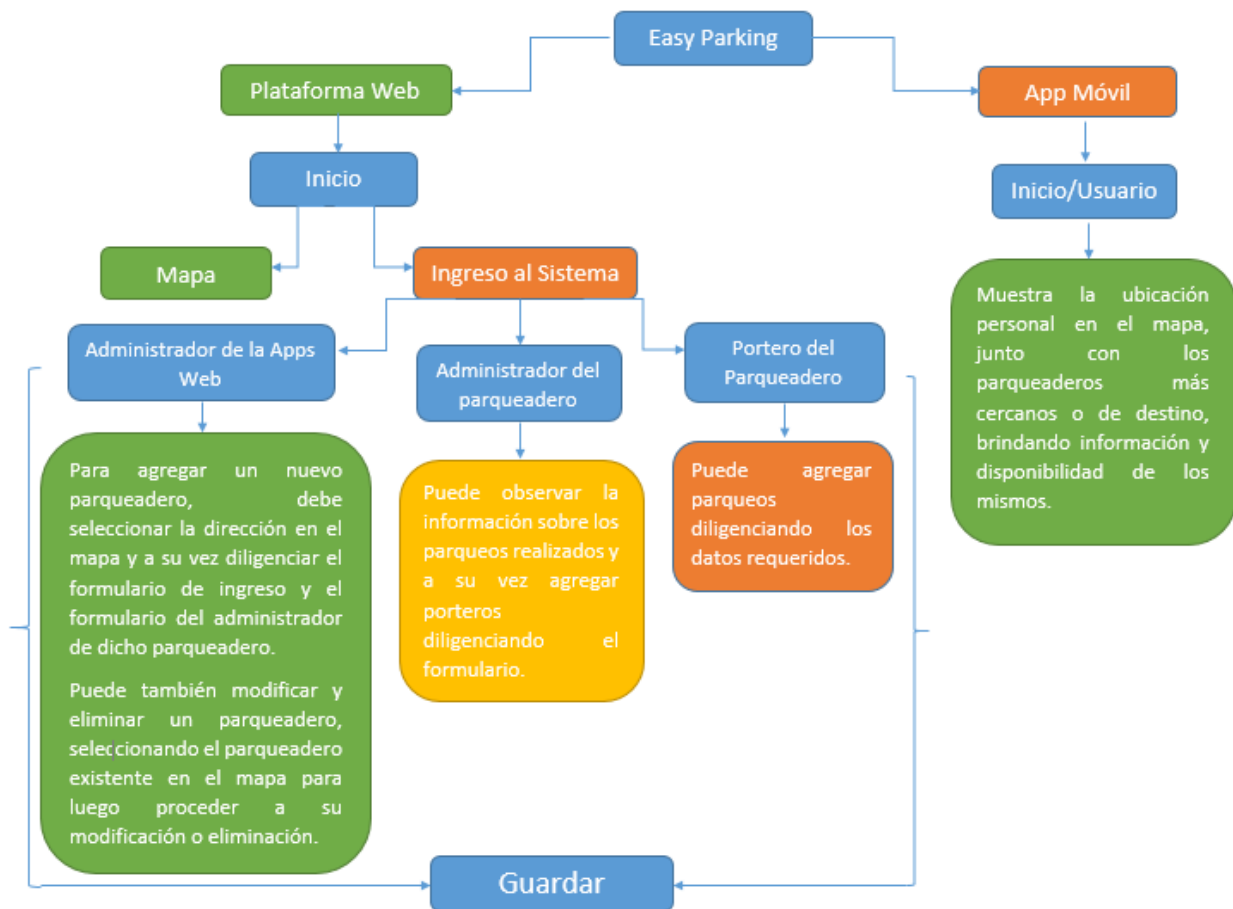
La Shield Ethernet va conectada por medio de un cable de red al computador en el cual se encuentra la base de datos que guarda la información de la disponibilidad para así mostrarla en la aplicación.

### 6.1.2. ARQUITECTURA DEL DISPOSITIVO Y EL SISTEMA



**Ilustración 7:** Arquitectura del dispositivo y el sistema, Fuente: Diseño Propio

## 6.2. DISEÑO DEL SISTEMA

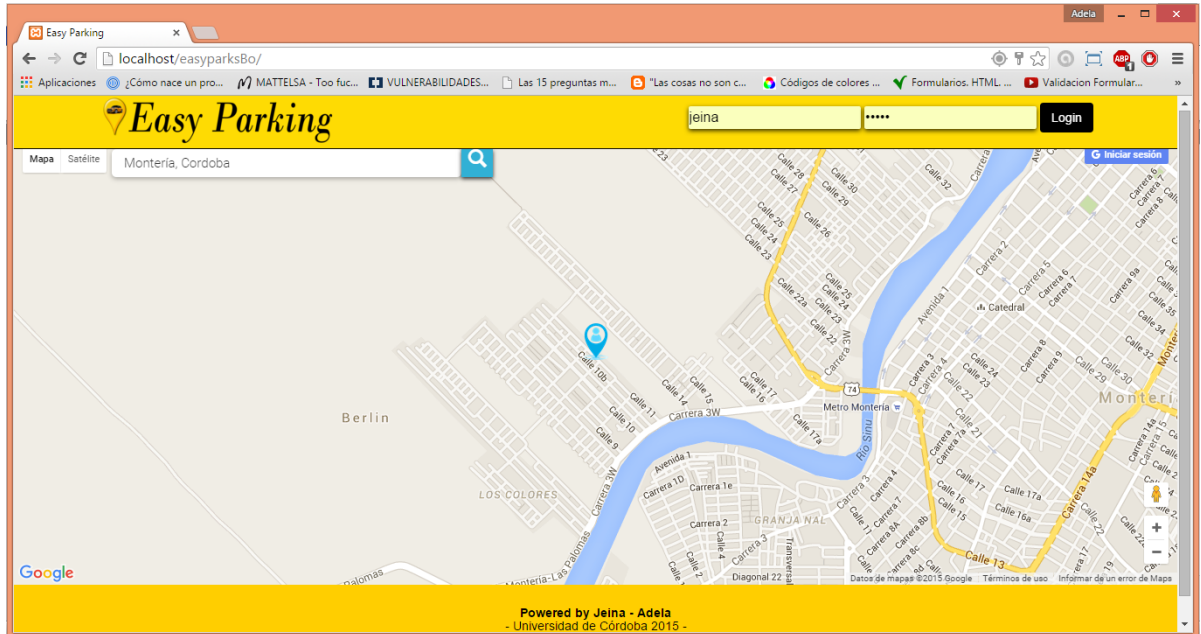


**Ilustración 8:** Diseño del sistema, Fuente: Diseño propio

## 6.2.1. INTERFAZ DEL SISTEMA

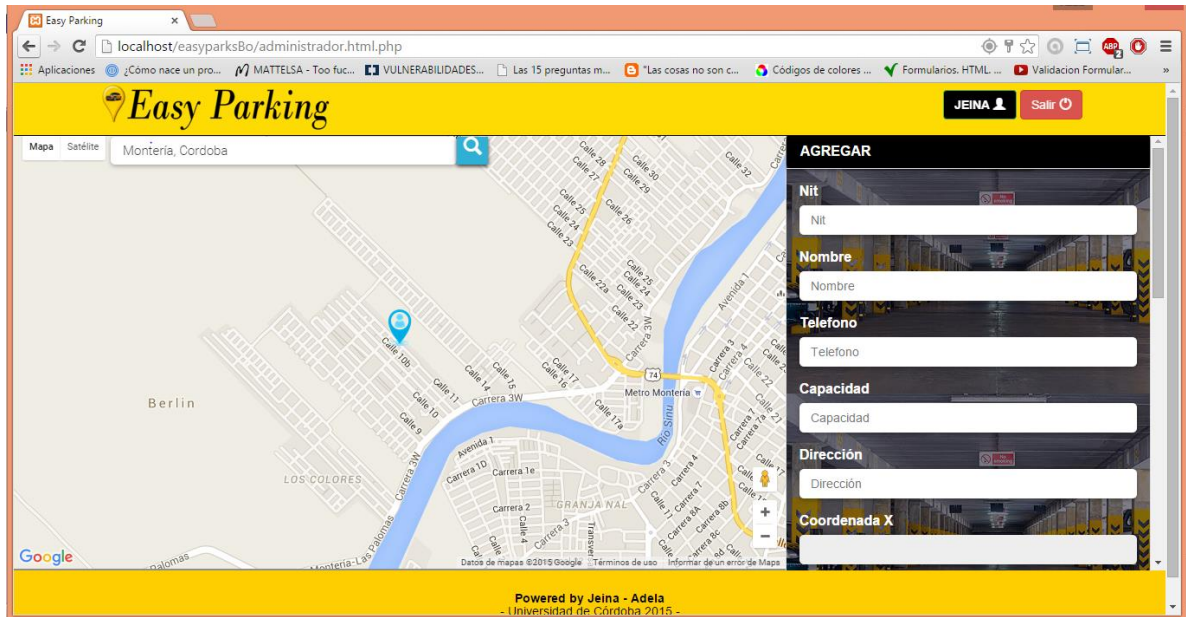
### 6.2.1.1. INTERFAZ DEL ADMINISTRADOR DE LA APPS WEBS

Interfaz de inicio de la Apps Web para el administrador de la Aplicación.



**Ilustración 9:** Interfaz de Inicio de la Apps Web para el administrador de la aplicación.

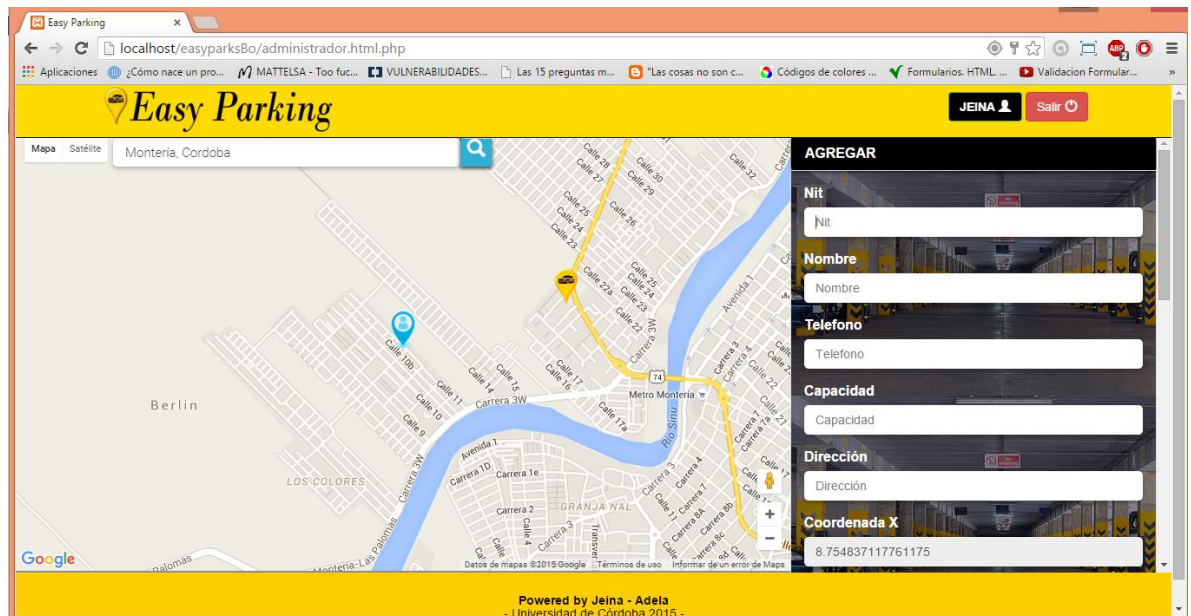
En esta interfaz se proporciona un Username y un Password para iniciar sesión como administrador de la aplicación, la labor de este administrador es agregar parqueaderos a la base de datos los cuales saldrán en la interfaz que vera el usuario.



**Ilustración 10:** Primera Interfaz del administrador de la aplicación

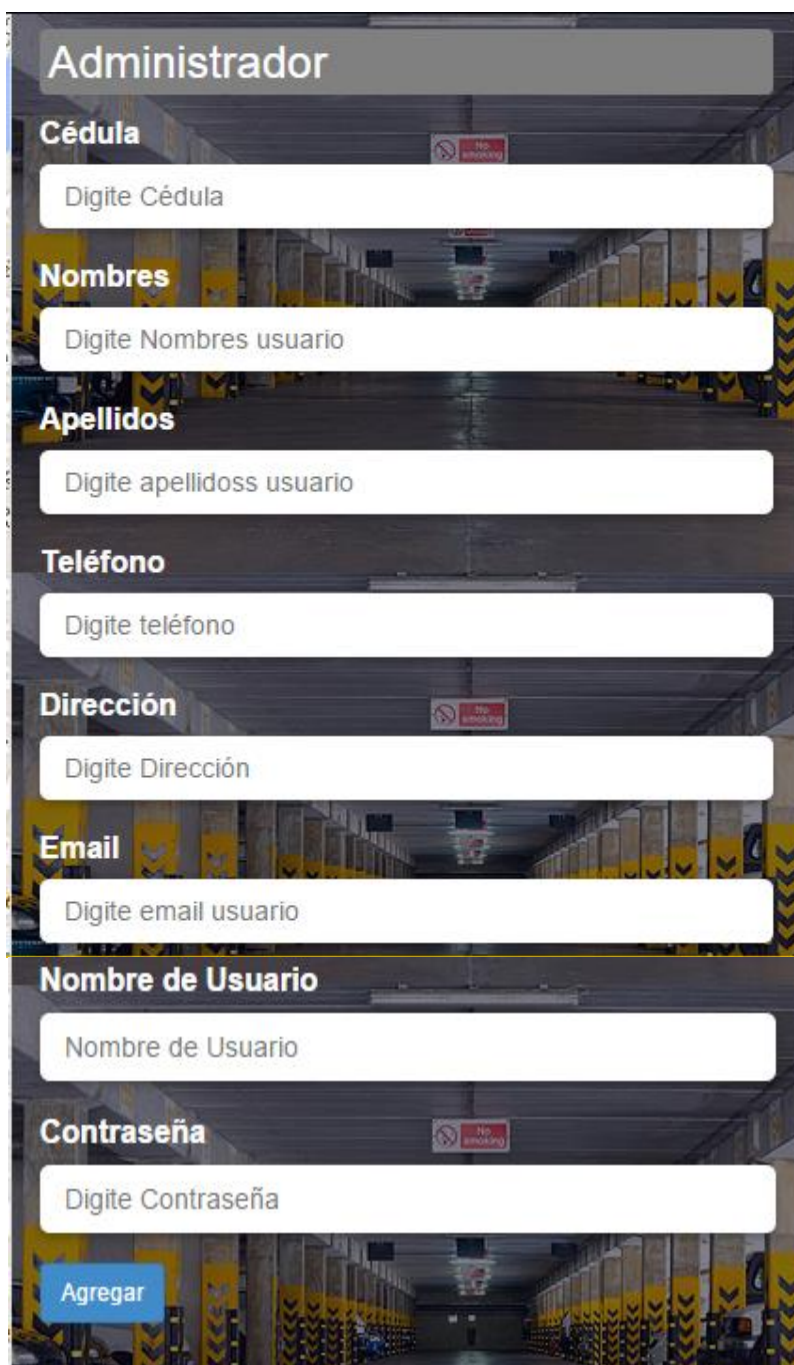
Para agregar nuevos parqueaderos se realizan los siguientes pasos:

1. Buscar la dirección del nuevo parqueadero en el mapa y hacer clic para agregar el marcador amarillo.



**Ilustración 11:** Interfaz del administrador cuando agrega el marcador del nuevo parqueadero.

Al agregar el nuevo marcador notamos que los datos de las coordenadas X y Y se insertan en el formulario automáticamente y nos da la opción de seguir llenando los campos inclusive los del administrador del nuevo parqueadero.



**Administrador**

**Cédula**  
Digite Cédula

**Nombres**  
Digite Nombres usuario

**Apellidos**  
Digite apellidos usuario

**Teléfono**  
Digite teléfono

**Dirección**  
Digite Dirección

**Email**  
Digite email usuario

**Nombre de Usuario**  
Nombre de Usuario

**Contraseña**  
Digite Contraseña

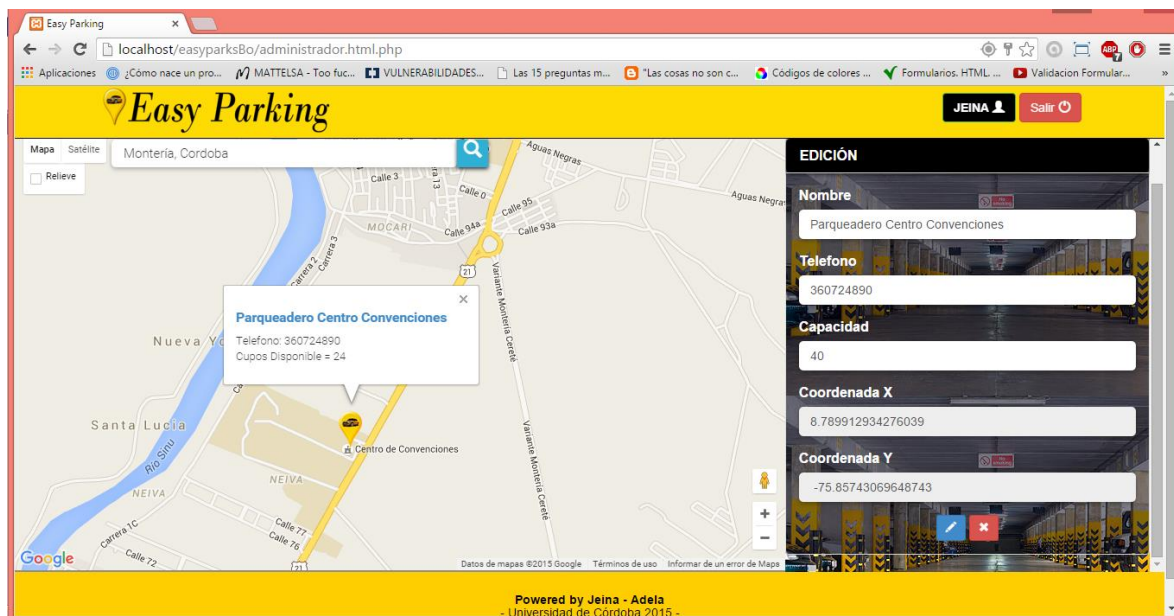
**Agregar**

**Ilustración 12:** Interfaz del formulario para insertar los datos del administrador del parqueadero nuevo.



Al llenar todos los campos del formulario y dar en el botón agregar se inserta un nuevo parqueadero en la base de datos.

Para Editar un parqueadero de la Base de datos de la aplicación se debe buscar el parqueadero y seleccionarlo.

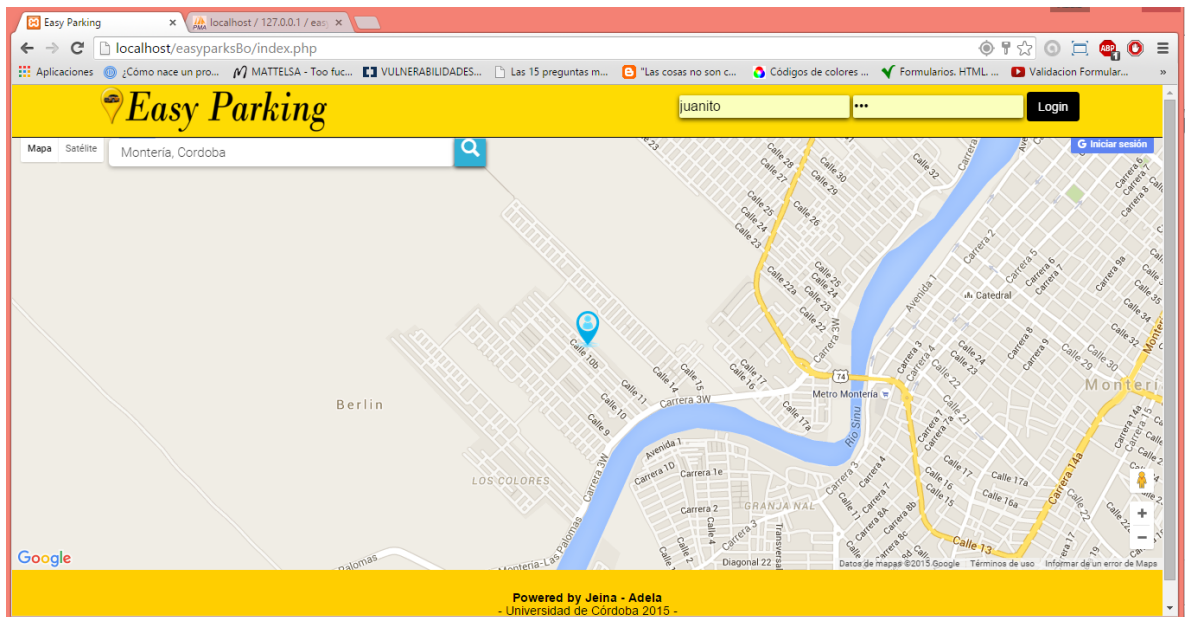


**Ilustración 13:** Interfaz del administrador de la Apps Web cuando edita un parqueadero.

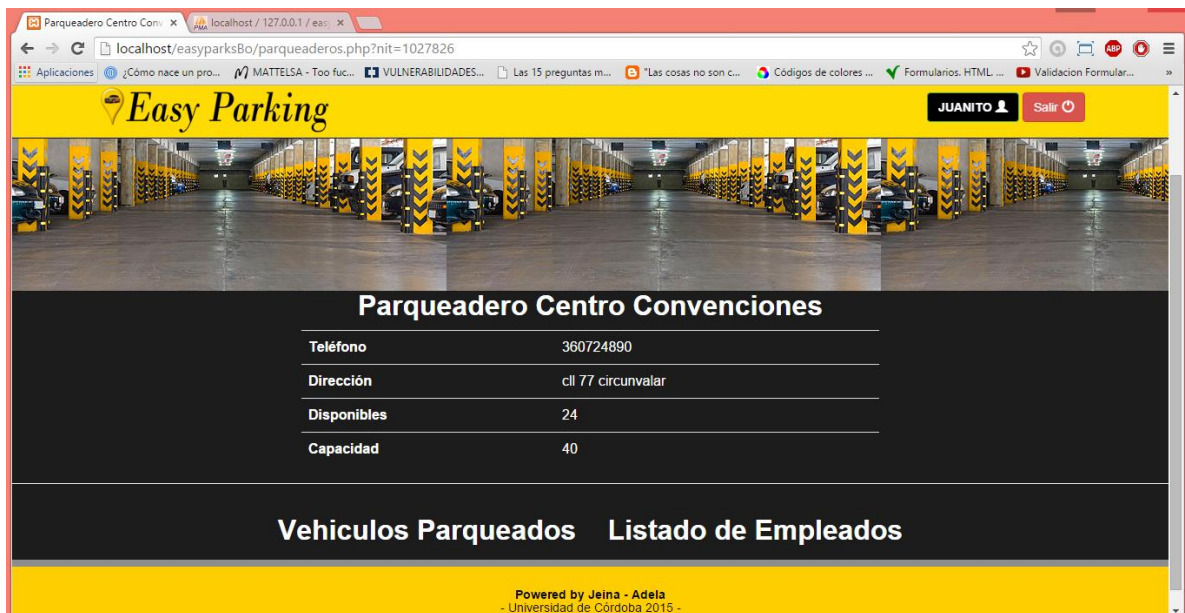
En la Edición de un parqueadero sale un formulario con los campos que se pueden editar (los que son de color blanco), se cambian los datos a editar y se le da actualizar. Pero si se desea borrar el parqueadero se le da en el botón Borrar.

## 6.2.2. INTERFAZ DEL ADMINISTRADOR DEL PARQUEADERO

Interfaz de inicio de la Apps Web para el administrador del parqueadero.



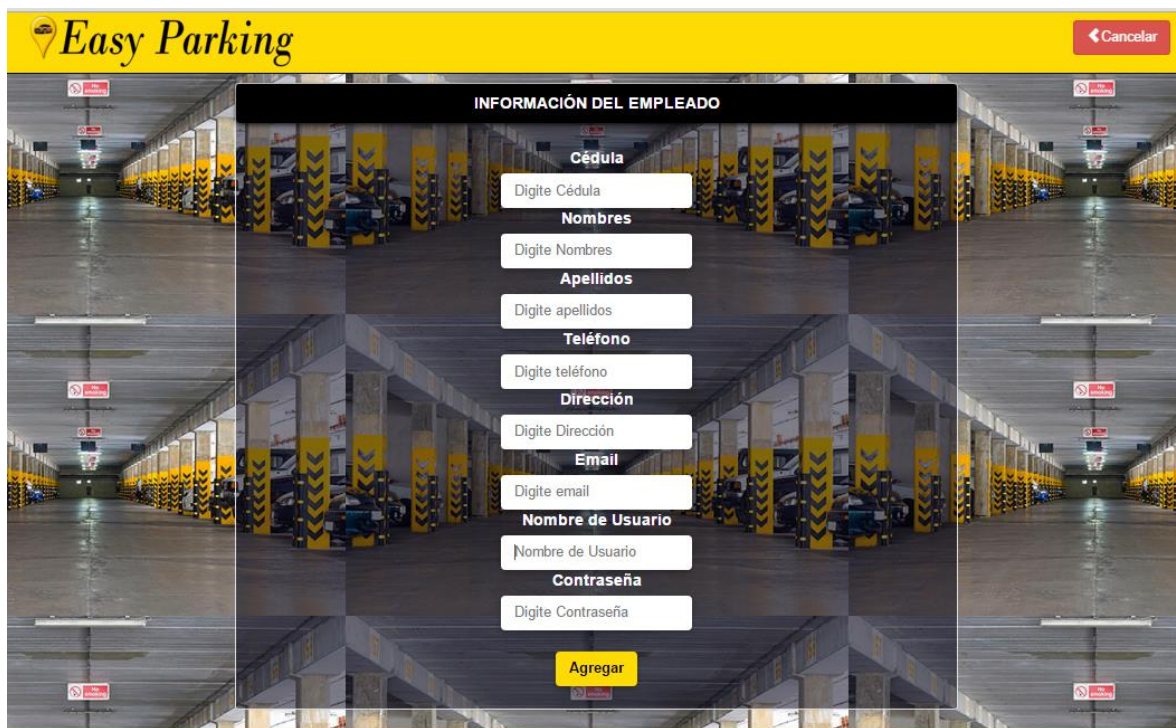
**Ilustración 14:** Interfaz del administrador del parqueadero cuando ingresa sus datos.



**Ilustración 15:** Primera Interfaz del administrador del parqueadero cuando inicia sesión.

El administrador del parqueadero en esta interfaz observa la información del parqueadero, de los parqueos que se han realizado y la de los porteros que están a su cargo.

Además tiene la opción de agregar más porteros con solo dar clic en el botón Agregar parqueo.



The screenshot displays the 'Easy Parking' application interface. At the top, there is a yellow header bar with the 'Easy Parking' logo on the left and a red 'Cancelar' button on the right. The main content area features a background image of a parking garage. Overlaid on this is a dark gray modal window titled 'INFORMACIÓN DEL EMPLEADO'. Inside this modal, there are several input fields, each preceded by a label: 'Cédula' (with a sub-label 'Digite Cédula'), 'Nombres' (with a sub-label 'Digite Nombres'), 'Apellidos' (with a sub-label 'Digite apellidos'), 'Teléfono' (with a sub-label 'Digite teléfono'), 'Dirección' (with a sub-label 'Digite Dirección'), 'Email' (with a sub-label 'Digite email'), 'Nombre de Usuario' (with a sub-label 'Nombre de Usuario'), and 'Contraseña' (with a sub-label 'Digite Contraseña'). At the bottom of the modal is a yellow 'Agregar' button.

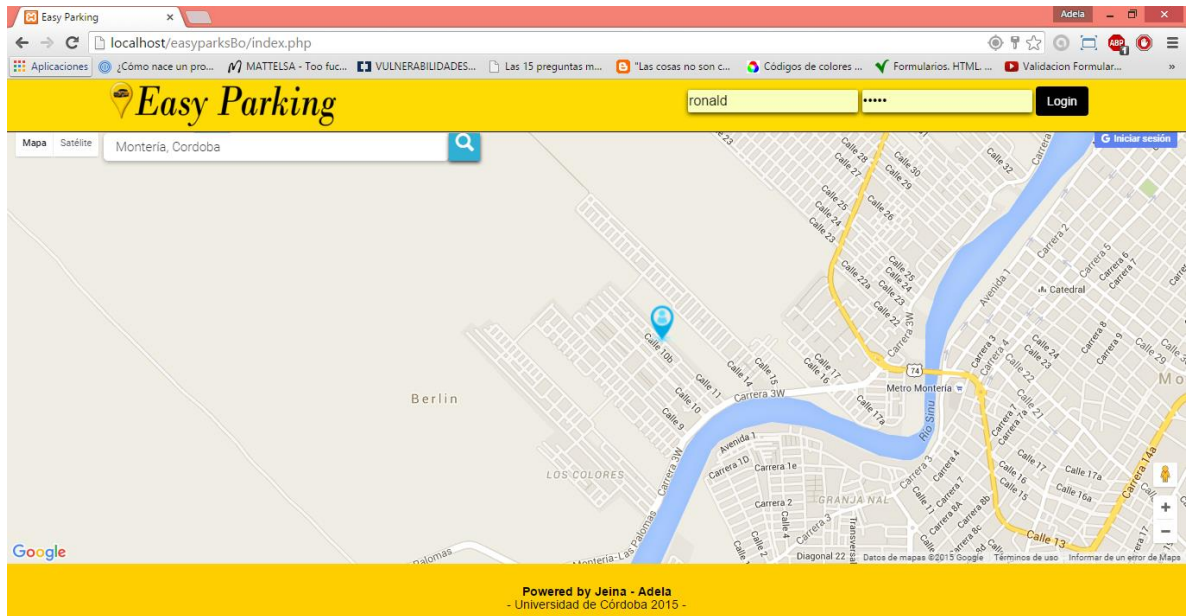
**Ilustración 16:** Interfaz del formulario para ingresar un nuevo portero.

Se debe diligenciar el formulario con todos los datos personales del portero además de crear sus datos para que pueda acceder a la aplicación Web.

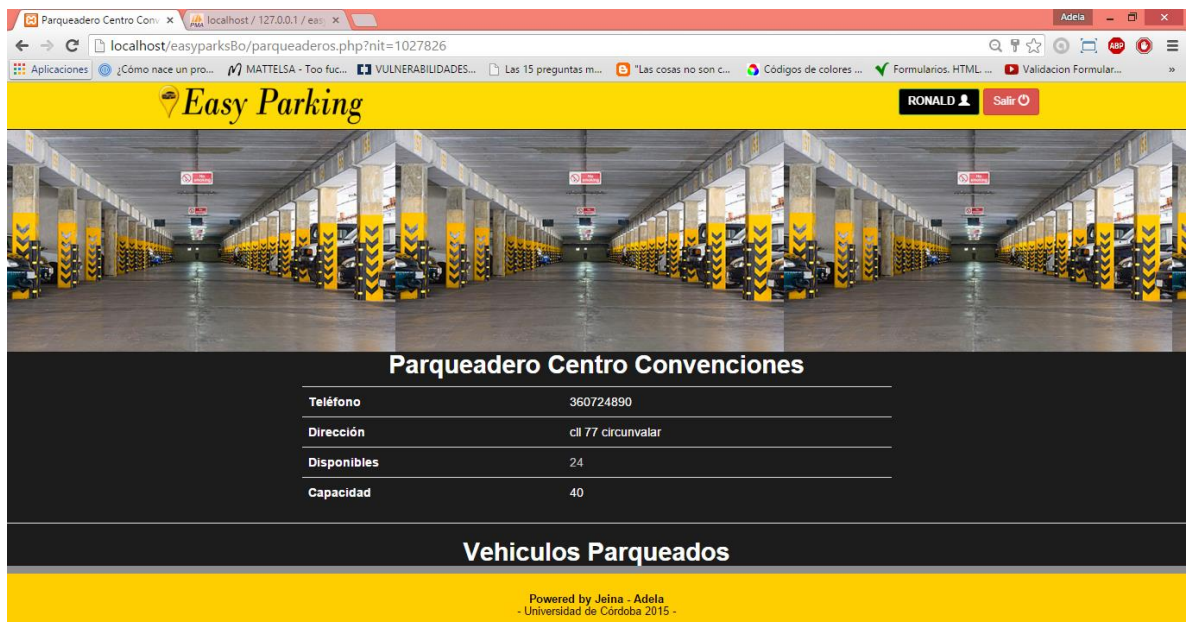


### 6.2.3. INTERFAZ DEL PORTERO DEL PARQUEADERO

Interfaz de inicio de la Apps Web para el portero del parqueadero.



**Ilustración 17:** Interfaz del portero del parqueadero cuando inicia sesión.



**Ilustración 18:** Primera Interfaz del portero del parqueaderos después de iniciar sesión.

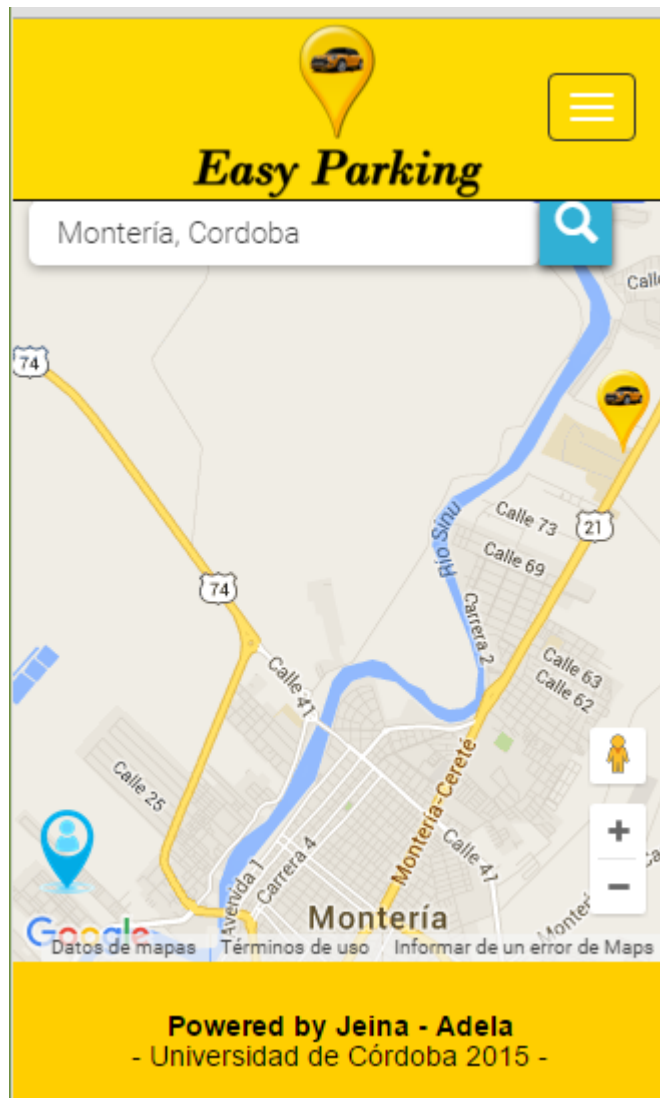
Para agregar un nuevo parqueo se le da clic en el botón Agregar Parqueo.

The screenshot displays the 'Easy Parking' application interface. At the top, there is a yellow header bar with the 'Easy Parking' logo on the left and a red 'Cancelar' button on the right. The main content area features a background image of a parking garage. Overlaid on this is a black modal box titled 'INFORMACIÓN DEL VEHÍCULO'. Inside this modal, there are four white input fields: 'Placa' (with a sub-label 'Digite Placa'), 'Color' (with a sub-label 'Digite Color'), and 'Tipo de Vehículo' (with a sub-label 'Digite Tipo'). Below these fields is a yellow 'Agregar' button. At the bottom of the application, a yellow footer bar contains the text 'Powered by Jeina - Adela - Universidad de Córdoba 2015 -'.

**Ilustración 19:** Interfaz del formulario que llena el portero del parqueadero para un nuevo parqueo.

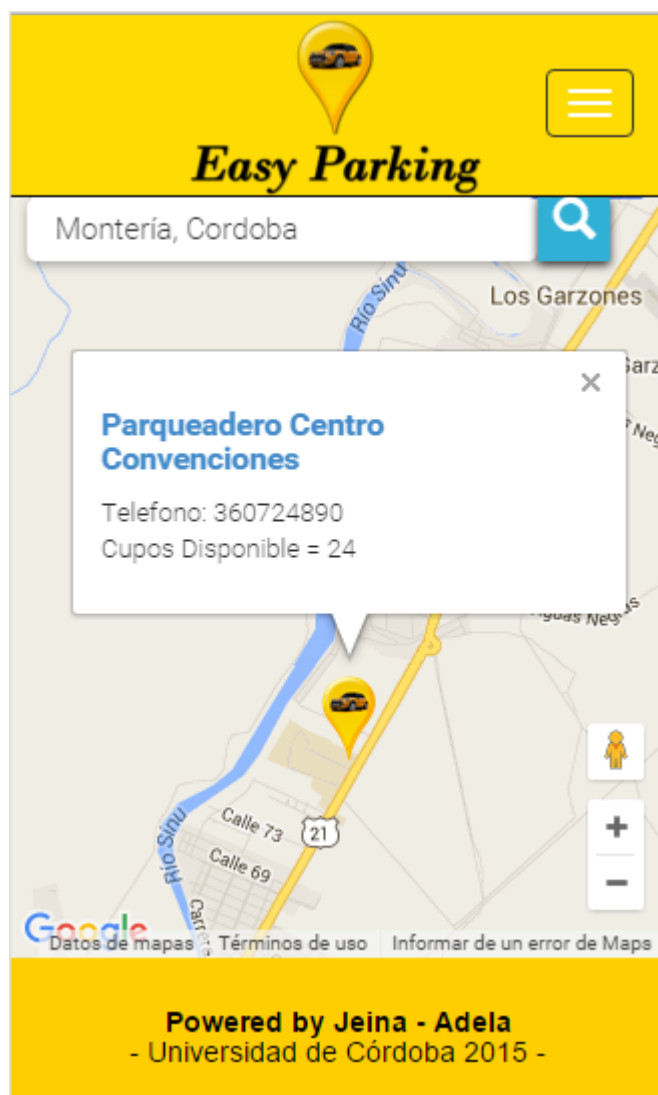
#### 6.2.4. INTERFAZ DEL USUARIO

Interfaz de inicio de la Apps Web para el usuario.



**Ilustración 20:** Interfaz de Inicio de la Apps Web

En esta interfaz encontramos un mapa que nos brinda la ubicación GPS del usuario (Puntero azul) que desea encontrar los parqueaderos que hacen parte de la base de datos de la aplicación (Puntero amarillo). Para saber la disponibilidad de un parqueadero en específico solo se debe buscar el puntero amarillo en el mapa y seleccionarlo.



**Ilustración 21:** Interfaz cuando el usuario escoge un parqueadero y desea ver su disponibilidad

En caso de que la información que se le proporciona en la interfaz anterior no sea suficiente y se desea saber más sobre el parqueadero se debe dar clic en el nombre del parqueadero, donde se le direccionara a la siguiente interfaz.



**Ilustración 22:** Interfaz cuando el usuario desea ver más información sobre un parqueadero en específico.

## 7. CONCLUSIONES

Mediante el desarrollo del proyecto, se logró obtener mayor conocimiento acerca de las nuevas tecnologías que están a la vanguardia, por cual esta aplicación representa una solución eficaz e innovadora para los parqueaderos de la ciudad de Montería del departamento de Córdoba.

De estos nuevos conocimientos adquiridos podemos concluir que las funciones proporcionadas por la API de Google Maps nos han permitido crear nuevas aplicaciones y personalizar los mapas digitales en gran medida, puesto que no sólo se limita a movernos por un mapa, sino que además nos permite obtener datos de suma importancia para los problemas que pretendemos resolver, como lo es saber la disponibilidad de los parqueaderos.

Dado los objetivos propuestos se menciona que se cumplieron a cabalidad, creando y partiendo que el sistema es totalmente funcional para los parqueaderos de la ciudad de Montería, especialmente en el centro de la misma, además la geolocalización obtenida de los parqueaderos genera a los usuarios su disponibilidad, lo cual es el objetivo principal de este proyecto.

Este sistema está constituido por dos partes, una parte web la cual cumple con la función de administrar el sistema y una aplicación móvil para los usuarios. Una de sus ventajas es la disminución de tiempo al momento de buscar un parqueadero con disponibilidad y también la facilidad de poder realizar la búsqueda desde cualquier lugar sin necesidad de estar presente en dicho parqueadero y ciudad.

## BIBLIOGRAFÍA

ABC, D. (s.f.). *definicionabc.com*. Obtenido de definicionabc.com:

<http://www.definicionabc.com/geografia/geolocalizacion.php>

Alvarez, M. A. (1 de Enero de 2001). *Desarrolloweb.com*. Obtenido de

Desarrolloweb.com: <http://www.desarrolloweb.com/articulos/que-es-html.html>

Bahit, E. (2 de Noviembre de 2013). *filtralub.com*. Recuperado el 4 de Octubre de 2015, de

filtralub.com:

<http://www.filtralub.com/admin/cursos/f94c96a6d573310f66e591ac5cb0757e.pdf>

Balcells J., R. J. (s.f.). *Sensores y Actuadores*. Obtenido de Sensores y Actuadores:

<http://isa.uniovi.es/docencia/autom3m/Temas/Tema7.pdf>

Cart, Z. (2015). *Arduino*. Obtenido de Arduino: <https://store.arduino.cc/product/A000066>

Chávez, M. E. (03 de 10 de 2011). *Escritorio del docente*. Obtenido de Escritorio del

docente:

[http://escritoriocentros.educ.ar/datos/Introduccion\\_geolocalizacion\\_google\\_earth.](http://escritoriocentros.educ.ar/datos/Introduccion_geolocalizacion_google_earth.html)

html

cmi. (s.f.). *cmi*. Recuperado el 8 de junio de 2015, de cmi: [www.cmi.com.co/parking-app:-](http://www.cmi.com.co/parking-app:-la-aplicacion-para-encontrar-parqueadero/227813)

[la-aplicacion-para-encontrar-parqueadero/227813](http://www.cmi.com.co/parking-app:-la-aplicacion-para-encontrar-parqueadero/227813)

*cscjprofes.com*. (febrero de 2014). Obtenido de cscjprofes.com: [http://cscjprofes.com/wp-](http://cscjprofes.com/wp-content/uploads/2014/02/teoria_arduino2009.pdf)

[content/uploads/2014/02/teoria\\_arduino2009.pdf](http://cscjprofes.com/wp-content/uploads/2014/02/teoria_arduino2009.pdf)

Developers. (24 de 10 de 2013). *Developers*. Obtenido de Developers:

<https://developers.google.com/maps/documentation/distancematrix/?hl=es>

EcuRed. (29 de 10 de 2014). *EcuRed*. Obtenido de EcuRed:

[http://www.ecured.cu/index.php/Sistema\\_Gestor\\_de\\_Base\\_de\\_Datos](http://www.ecured.cu/index.php/Sistema_Gestor_de_Base_de_Datos)

Google, I. d. (s.f.). Obtenido de

<https://www.google.com/search?q=arduino&espv=2&biw=1366&bih=667&site=webhp&source=lnms&tbn=isch&sa=X&ved=0CAcQAUoAmoVChMIjNaQjPjqxwIVBdceCh3YxAri#tbn=isch&q=tipod+arduino&imgsrc=tSOP-948XeK4BM%3A>

html.net. (18 de 9 de 2013). *HTML.NET*. Obtenido de HTML.NET:

<http://es.html.net/tutorials/css/lesson1.php>

Ibañez. (02 de 10 de 2013). *Sinbad2*. Obtenido de Sinbad2:

[http://sinbad2.ujaen.es/cod/archivosPublicos/pfc/pfc\\_antonio\\_araque.pdf](http://sinbad2.ujaen.es/cod/archivosPublicos/pfc/pfc_antonio_araque.pdf)

Ibañez, A. (15 de Septiembre de 2008). *Sinbad2*. Obtenido de Sinbad2:

[http://sinbad2.ujaen.es/cod/archivosPublicos/pfc/pfc\\_antonio\\_araque.pdf](http://sinbad2.ujaen.es/cod/archivosPublicos/pfc/pfc_antonio_araque.pdf)

Khaled Abou, A., & Joshua. (7 de 04 de 2013). *NOTICIAS GOOGLE MAPS*. Obtenido de

NOTICIAS GOOGLE MAPS: [http://www.googlemaps.es/?page\\_id=3](http://www.googlemaps.es/?page_id=3)

maestrosdelweb.com. (23 de mayo de 2001). *maestrosdelweb.com*. Obtenido de

maestrosdelweb.com: <http://www.maestrosdelweb.com/editorial/phpintro/>

Martinez Orozco, A. (28 de Febrero de 2013). *blogspot.com*. Obtenido de blogspot.com:

<http://gestoresadrian.blogspot.com/>

Pujol Perez, J. (21 de febrero de 2013). *SlideShare*. Obtenido de SlideShare:

[es.slideshare.net/josepujolpherez/ethernet-shield-16681047](http://es.slideshare.net/josepujolpherez/ethernet-shield-16681047)

QODE. (17 de abril de 2014). *qode.pro*. Obtenido de qode.pro: [http://qode.pro/blog/que-es-](http://qode.pro/blog/que-es-una-web-app/)

[una-web-app/](http://qode.pro/blog/que-es-una-web-app/)

Rodriguez, E. (30 de Enero de 2012). *SlideShare*. Obtenido de SlideShare:

[es.slideshare.net/efelixrdz/sensores-fotoelctricos](http://es.slideshare.net/efelixrdz/sensores-fotoelctricos)

Tasigchana Villegas, J. G. (19 de Marzo de 2014). *Repositorio.ucsg.edu*. Recuperado el 25

de Junio de 2015, de Repositorio.ucsg.edu.:



<http://repositorio.ucsg.edu.ec:8080/bitstream/123456789/1241/1/T-UCSG-PRE-ING-CIS-66.pdf>

Viracachá, M. (2012). *eprints.rclis.org*. Recuperado el 2 de julio de 2015, de

*eprints.rclis.org*:

<http://eprints.rclis.org/17739/1/Dise%C3%B1o%20de%20un%20Sistema%20de%20Gesti%C3%B3n%20del%20Conocimiento%20para%20el%20Sistema%20de%20Bibliotecas%20de%20la%20Universidad%20de%20los%20Andes.pdf>

Wikipedia.org. (2 de septiembre de 2015). *Wikipedia.org*. Obtenido de Wikipedia.org:

[http://es.wikipedia.org/wiki/Sistema\\_de\\_posicionamiento\\_global](http://es.wikipedia.org/wiki/Sistema_de_posicionamiento_global)

# **ANEXO**

## **ANEXO**

### **5.1. ESPECIFICACIONES DE REQUISITOS**

En esta unidad realizaremos una descripción completa del comportamiento del sistema que se va a desarrollar. Incluye un conjunto de requisitos funcionales (definen el comportamiento interno del software). Además de los requisitos funcionales, también contienen los requisitos no funcionales (o complementarios) que son requisitos que imponen restricciones en el diseño o la implementación.

#### **✓ ESPECIFICACION DE REQUISITOS**

- Objetivos del sistema.
- Requisitos de información.
- Restricciones de información.

#### **✓ REQUISITOS FUNCIONALES.**

- Diagramas de casos de usos.
- Definición de actores.
- Documentación de los casos de usos.

#### **✓ REQUISITOS NO FUNCIONALES.**

##### **5.1.1. OBJETIVOS DEL SISTEMA**

A continuación se muestra una tabla en la que se detallan los objetivos que se alcanzaron con el desarrollo del sistema, el cual se encuentra disponible para toda la comunidad interesada en utilizar sus servicios.

<b>ID OBJETIVO</b>	<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>	<b>ESTABILIDAD</b>
<b>01</b>	Control de acceso	El sistema Easy Parking permitirá el acceso al usuario (sin ser registrado), administrador de la Apps Web, administrador del parqueadero y porteros del mismo, siempre y cuando estos si se encuentren registrados.	Alta
<b>02</b>	Sistema de GPS	El sistema realizara en tiempo real por medio de la api de Google Maps un rastreo de los parqueaderos registrados y mostrara su información, especialmente la de la disponibilidad.	
<b>03</b>	Administración de parqueaderos	El sistema permitirá administrar, ingresar, consultar, modificar, eliminar toda la información referente a los parqueaderos que se encuentren en la base de datos.	Alta
<b>04</b>	Navegación	El sistema Easy Parking permitirá a los usuarios sin necesidad de registro visualizar los parqueaderos que ofrece la Apps Web.	Alta
<b>05</b>	Búsqueda	El sistema permitirá a los usuarios que se familiarice con documentación	Alta

		acerca de los diferentes parqueaderos que se manejaran en la Apps Web.	
<b>06</b>	Administración de porteros	El sistema Easy Parking permitirá al administrador del parqueadero acceder a la información personal y al historial de parqueo de cada portero.	Alta
<b>07</b>	Estadísticas del sistema	El sistema permitirá obtener información relacionada con el uso del sistema, la cantidad de puestos disponibles, los usuarios que han accedido a ellos, los porteros que han realizado el parqueo.	Alta

**Tabla 5:** Objetivos del sistema.

### 5.1.1. REQUISITOS DE INFORMACIÓN

Se definen los requerimientos concernientes al procesamiento de la información que será almacenada en el sistema, los cuales se definieron para el desarrollo del mismo. Se especificará toda la información que se debe almacenar en el sistema, mediante las siguientes tablas.

<b>RI- 01</b>	<b>INFORMACIÓN SOBRE CONTROL DE ACCESO.</b>
<b>Objetivos asociados</b>	OBJETIVO 01-Control de Acceso
<b>Requisitos asociados</b>	<b>RF-01</b> Controlar accesos de usuarios como administrador de la Apps Web, administrador del parqueadero y porteros.
<b>Descripción</b>	El sistema Easy Parking deberá permitir controlar el acceso de los diferentes tipos

	de usuarios que tiene el sistema.
<b>Datos específicos</b>	Nombre de usuario y contraseña (username y password), para el administrador de la Apps Web, administrador del parqueadero y porteros. El usuario común no lo necesitara.
<b>Tiempo de vida promedio</b>	5 años
<b>Ocurrencias</b>	
<b>Comentarios</b>	Ninguno.

**Tabla 6:** Requisitos de información control de acceso.

<b>RI- 02</b>	<b>INFORMACIÓN SOBRE SISTEMA DE GPS.</b>
<b>Objetivos asociados</b>	OBJETIVO 02- Sistema de GPS.
<b>Requisitos asociados</b>	<b>RF-02</b> Registrar Coordenadas. <b>RF-03</b> Actualizar Coordenada. <b>RF-04</b> Graficar Coordenadas. <b>RF-05</b> Consultar Coordenadas.
<b>Descripción</b>	El sistema Easy Parking rastreara los parqueaderos que se encuentren registrados, y mostrara su disponibilidad.
<b>Datos específicos</b>	Identificación, longitud, descripción, nombre, dirección, disponibilidad.
<b>Tiempo de vida promedio</b>	5 años
<b>Ocurrencias</b>	
<b>Comentarios</b>	Ninguno.

**Tabla 7:** Requisitos de información Sistema de GPS.

<b>RI- 03</b>	<b>INFORMACIÓN SOBRE ADMINISTRACIÓN DE PARQUEADEROS</b>
<b>Objetivos asociados</b>	OBJETIVO 03- Administración de parqueaderos.
<b>Requisitos asociados</b>	<b>RF-06</b> Agregar parqueadero en Easy Parking. <b>RF-07</b> Modificar parqueaderos en Easy Parking. <b>RF-08</b> Eliminar parqueaderos en Easy Parking.

<b>Descripción</b>	El sistema Easy Parking deberá permitir almacenar la información de los diferentes parqueaderos del sistema.
<b>Datos específicos</b>	Nombre, dirección, teléfono, capacidad, cupos disponibles y coordenadas X y Y.
<b>Tiempo de vida promedio</b>	5 años
<b>Ocurrencias</b>	
<b>Comentarios</b>	Ninguno.

**Tabla 8:** Requisitos de información administración de productos.

<b>RI- 04</b>	<b>INFORMACIÓN SOBRE NAVEGACIÓN</b>
<b>Objetivos asociados</b>	OBJETIVO 04- Navegación.
<b>Requisitos asociados</b>	<b>RF-09</b> Desplazarse por las secciones del sistema. <b>RF-10</b> Visualizar parqueaderos registrados.
<b>Descripción</b>	El sistema Easy Parking brindará un mapa de navegación muy intuitivo, mediante el cual los usuarios podrán visualizar los diferentes parqueaderos registrados. El mapa de navegación permitirá a los usuarios saber su ubicación actual y la ubicación de los parqueaderos, permitiéndoles desplazarse a cualquier de ellos.
<b>Datos específicos</b>	Contador de puestos disponible por parqueadero registrado.
<b>Tiempo de vida promedio</b>	5 años
<b>Ocurrencias</b>	
<b>Comentarios</b>	Ninguno.

**Tabla 9:** Requisitos de información de navegación.

<b>RI- 05</b>	<b>INFORMACIÓN SOBRE BÚSQUEDA.</b>
<b>Objetivos asociados</b>	OBJETIVO 05- Búsqueda.
<b>Requisitos asociados</b>	<b>RF-11</b> Se utilizara para obtener información acerca de los parqueaderos solicitados por el usuario según sus necesidades y la disponibilidad de los mismos. <b>RF-12</b> Se garantiza que la búsqueda sea sencilla pero que los resultados sean los más eficientes posibles dependiendo de los criterios de la búsqueda.

	<b>RF-13</b> Dar a conocer al usuario toda la información pertinente a la búsqueda solicitada para que este quede satisfecho de los resultados.
<b>Descripción</b>	El sistema Easy Parking contara con búsquedas de mucha simplicidad para el usuario pero con gran validez en los resultados.
<b>Datos específicos</b>	<ul style="list-style-type: none"> <li>▪ Búsqueda</li> <li>- Parqueaderos</li> </ul>
<b>Tiempo de vida promedio</b>	5 años
<b>Ocurrencias</b>	
<b>Comentarios</b>	Ninguno.

**Tabla 10:** Requisitos de información Búsqueda

<b>RI- 06</b>	<b>INFORMACIÓN SOBRE ADMINISTRACIÓN DE PORTEROS.</b>
<b>Objetivos asociados</b>	OBJETIVO 06- Administración de porteros.
<b>Requisitos asociados</b>	<b>RF-14</b> Consultar parqueos efectuados <b>RF-15</b> Agrega parqueos
<b>Descripción</b>	Easy Parking permitirá a los administradores consultar los parqueos efectuados por los porteros de cada parqueadero del sistema.
<b>Datos específicos</b>	<ul style="list-style-type: none"> <li>▪ Administración de porteros (Ver y Agregar)</li> <li>▪ Roles</li> </ul>
<b>Tiempo de vida promedio</b>	5 años
<b>Ocurrencias</b>	
<b>Comentarios</b>	Ninguno.

**Tabla 11:** Requisitos de información administración de porteros.

<b>RI- 07</b>	<b>INFORMACIÓN SOBRE ESTADÍSTICAS DEL SISTEMA.</b>
<b>Objetivos asociados</b>	OBJETIVO 07- Estadísticas del sistema.
<b>Requisitos asociados</b>	<b>RF-16</b> Consultar parqueos en el sistema.
<b>Descripción</b>	Easy Parking permitirá a los administradores del sistema y a los administradores del parqueadero conocer datos de gran importancia relacionados con el sistema, como la cantidad de parqueaderos y la cantidad de parqueos efectuados.



<b>Datos específicos</b>	▪ Nit de los parqueaderos, parqueos registrados con sus respectivas fechas y horas y personal que realizo el registro.
<b>Tiempo de vida promedio</b>	5 años
<b>Ocurrencias</b>	
<b>Comentarios</b>	Ninguno.

**Tabla 12:** Requisitos de información de estadísticas del sistema.

## 5.2. REQUISITOS FUNCIONALES

Requerimientos Funcionales Definen las funciones que el sistema será capaz de realizar. Describen las transformaciones que el sistema realiza sobre las entradas para producir salidas.

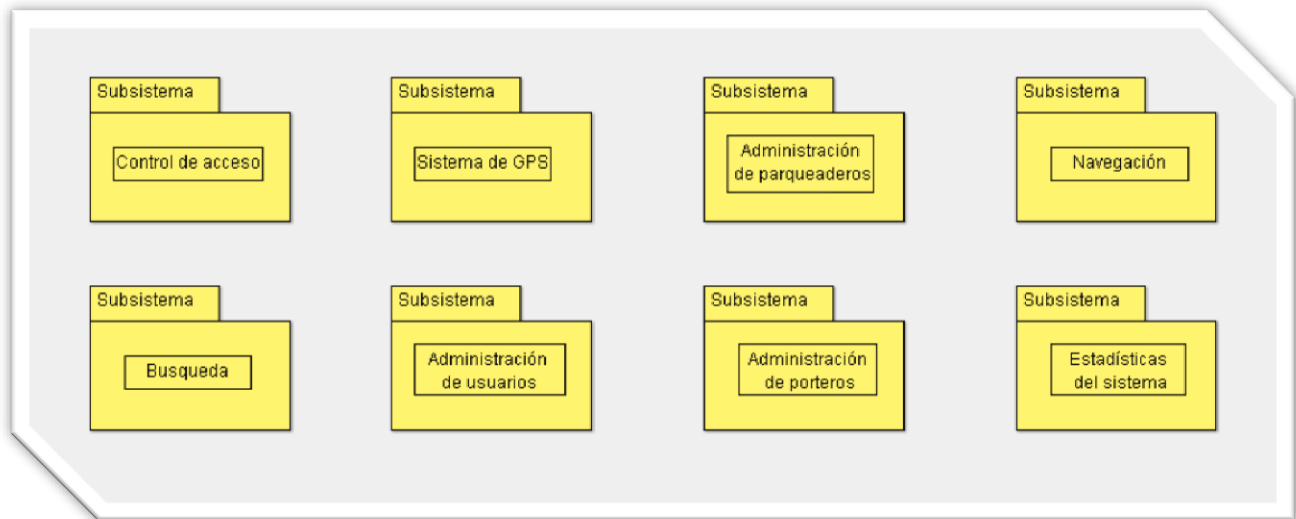
En este apartado se explican los requisitos funcionales del sistema Easy Parking en él encontraremos los diagramas de casa de uso, definición de actores y la documentación de los casos de uso.

### 5.2.1. DIAGRAMAS DE CASOS DE USO

Los diagramas de casos de uso documentan el comportamiento de un sistema desde el punto de vista del usuario, es decir, se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en el mismo. Por lo tanto los casos de uso determinan los requisitos funcionales del sistema, es decir, representan las funciones que un sistema puede ejecutar.

Su ventaja principal es la facilidad para interpretarlos, lo que hace que sean especialmente útiles en la comunicación con el usuario.

El sistema Easy Parking estableció las estructuras de los diagramas de casos de uso de la siguiente manera: Modulo de usuario, módulo de administrador de Apps Web, módulo de



administrador del parqueadero y módulo de portero del parqueadero, los cuales se encuentran representados en los siguientes diagramas:

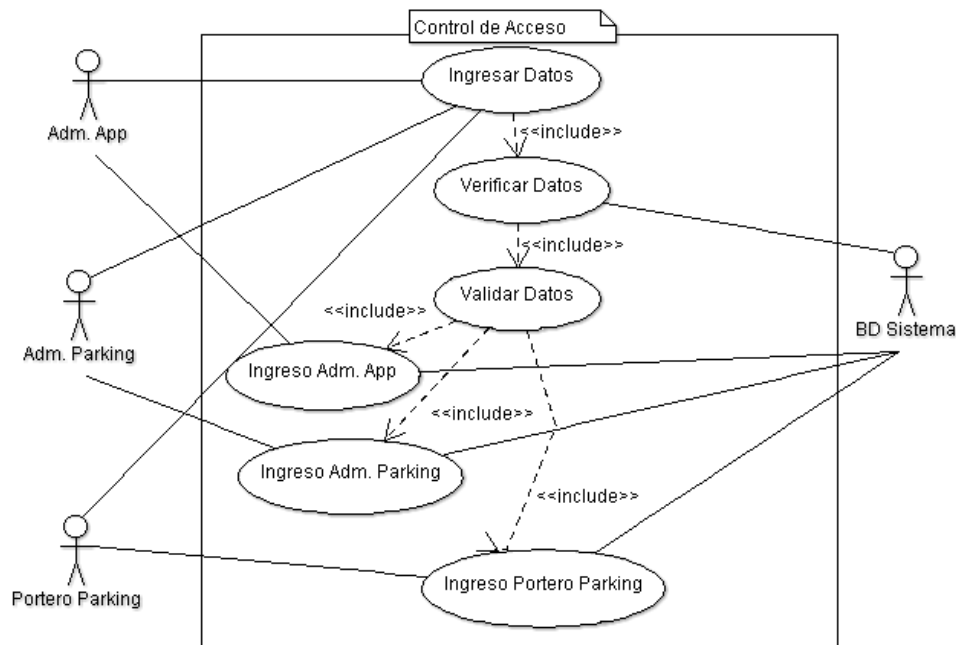
**Ilustración 23:** Diagrama de subsistemas.

#### **5.2.1.1. DIAGRAMA DE CASOS DE USO CONTROL DE ACCESO**

El diagrama de casos de uso de control de acceso, tal como se muestra en la figura **Diagrama casos de uso 01: control acceso**, orienta sobre las funcionalidades del sistema de cara a los autores, cuando estos tratan de entrar al sistema y esquematiza todas las operaciones realizadas tanto para el sistema, como por el autor y mantienen el siguiente orden:

- Ingresar Datos (Username y Password)
- Validar datos
- Conectar con la base de datos
- Verificar datos

- Ingreso de Adm. Apps Web
- Ingreso de administrador del parqueadero
- Ingreso del portero del parqueadero
- Entran al sistema de Base de datos



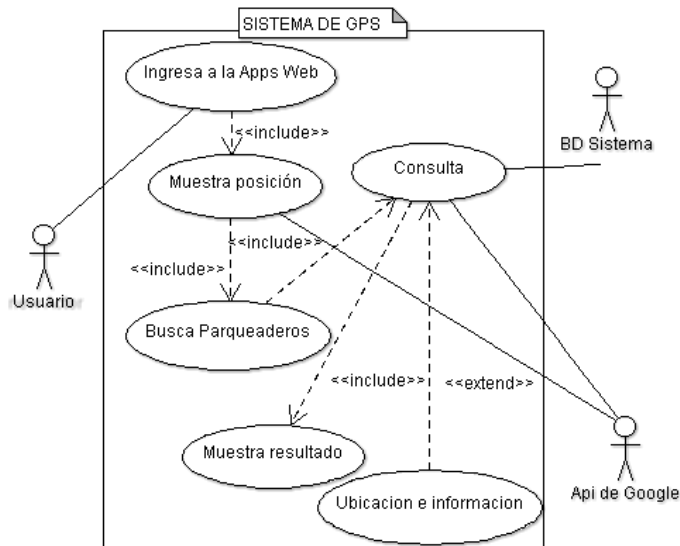
**Ilustración 24:** Diagrama casos de uso 01 Control Acceso

#### 5.1.1.1.DIAGRAMA DE CASOS DE USO SISTEMA DE GPS

El diagrama de casos de uso de sistema de posición o GPS, tal como se muestra en la figura

**Diagrama casos de uso 02: sistema de GPS**, orienta sobre las funcionalidades del sistema de cara con el usuario, cuando este trata de entrar al sistema y esquematiza todas las operaciones realizadas tanto para el sistema, como por el autor y mantienen el siguiente orden:

- |                           |                           |
|---------------------------|---------------------------|
| - Ingresar a la Apps Web. | - Consulta                |
| - Muestra posición        | - Ubicación e información |
| - Busca parqueaderos      | - Mostrar resultados      |

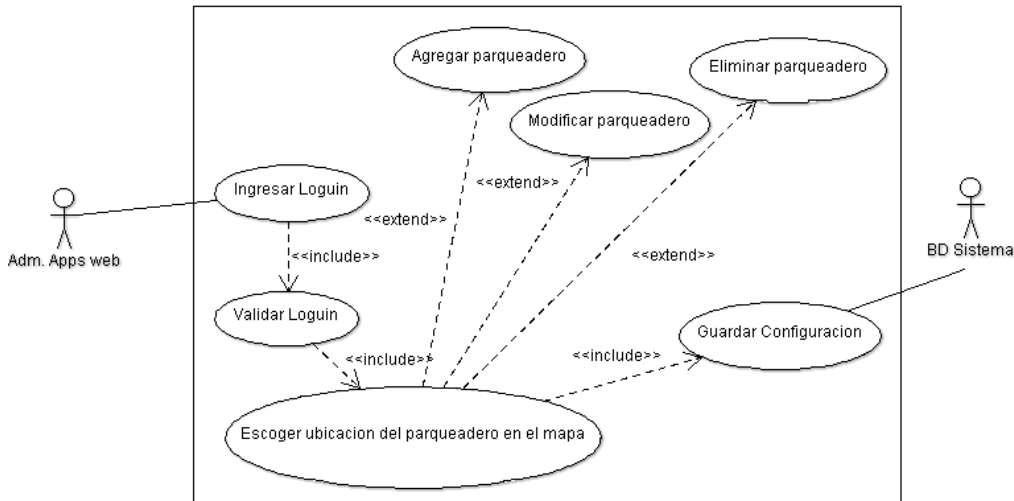


**Ilustración 25:** Diagrama casos de uso 02 Sistema de GPS

#### 5.1.1.2. DIAGRAMA DE CASO DE USO ADMINISTRACION DE PARQUEADEROS.

El diagrama de casos de usos de administración de parqueadero del sistema Easy Parking (OBJETIVO-03), tal como se muestra en la figura, **Diagrama de caso de uso 03: Administración de parqueaderos.** Nos modela como el administrador de la Apps Web puede configurar los parqueaderos que se verán en el sistema Easy Parking, esto permitirá que el sistema permanezca actualizado, además de poder eliminar los parqueaderos que ya no requieran el servicio. Las operaciones del administrador son las siguientes:

- Ingresar Loguin
- Validar Loguin
- Escoger ubicación del parqueadero en el mapa
  - Agregar parqueaderos
  - Modificar Parqueaderos
  - Eliminar Parqueaderos
- Guardar configuración



**Ilustración 26:** Diagrama de caso de uso 07 Administración de parqueadero.

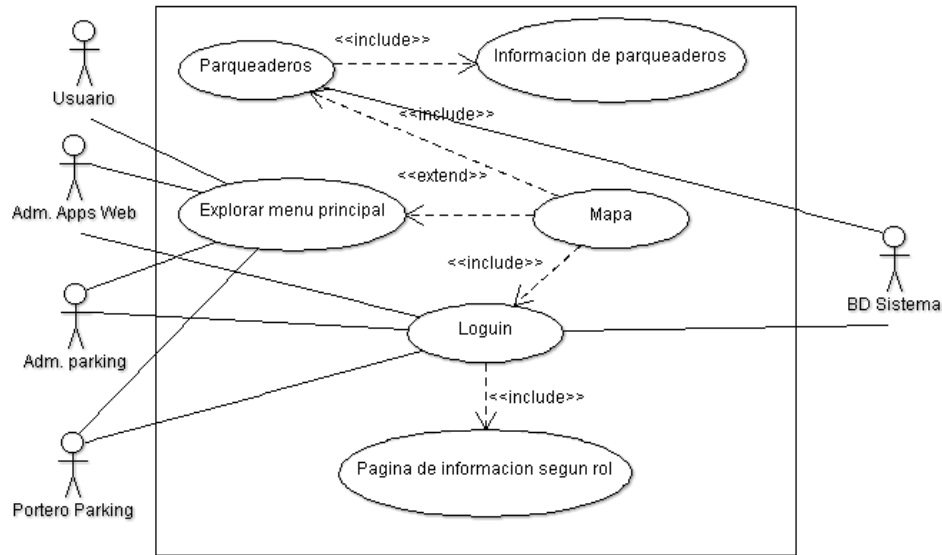
### 5.1.1.3. DIAGRAMA DE CASO DE USO NAVEGACIÓN

El sistema brinda un menú que permite la navegación y visualización de las diferentes secciones del sistema Easy Parking. Este diagrama está relacionado con el OBJETIVO 04 del sistema. Cabe mencionar que todos los usuarios (registrados y no registrados) podrán visualizar los parqueaderos, sin embargo solo los usuarios registrados (administradores y porteros) podrán realizar otras operaciones más avanzadas que se describen en el siguiente diagrama de caso de uso:

Las operaciones relacionadas con la navegación que podrán realizar los diferentes tipos de usuarios son:

- Explorar menú principal
- Mapa
- Parqueaderos

- Información de parqueaderos
- Login
- Página de información según rol.



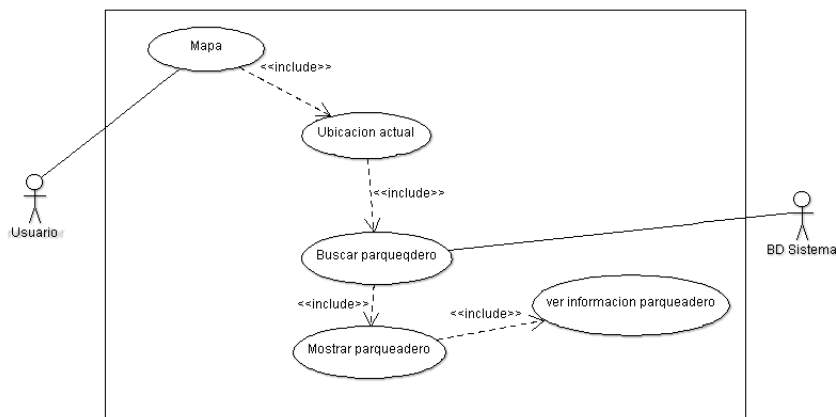
**Ilustración 27:** Diagrama de caso de uso 08: Navegación.

#### 5.1.1.4.DIAGRAMA DE CASO DE USO BUSQUEDA

El diagrama de casos de usos de Búsqueda, tal como se muestra en la figura **Diagrama de caso de uso 09: Búsqueda**, nos muestra como un usuario podrá buscar la información acerca de los parqueaderos y su disponibilidad.

Las operaciones del usuario son las siguientes:

- Mapa
- Ubicación actual
- Buscar parqueadero
- Mostrar parqueadero
- Ver información parqueadero



**Ilustración 28:** Diagrama de caso de uso 09: Búsqueda.

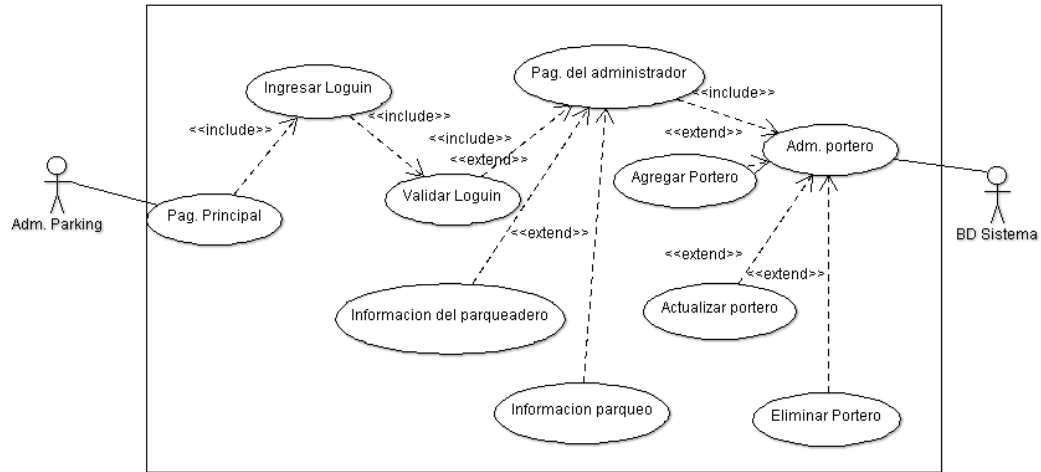
#### 5.1.1.5. DIAGRAMA DE CASO DE USO ADMINISTRACION DE PORTERO.

El diagrama de casos de usos de administración de portero del sistema Easy Parking (OBJETIVO-07), tal como se muestra en la figura, **Diagrama de caso de uso 11: Administración de portero.** Nos muestra como el administrador del sistema puede administrar los porteros que acceden al sistema, es decir, puede agregar, actualizar y eliminar porteros que estarán a cargo de los parqueos que se realicen.

Las opciones del administrador son las siguientes:

- Pagina principal
- Ingresar loguin.
- Validar loguin.
- Página del administrador
  - Información del parqueadero
  - Información parqueo
- Administrar portero.
  - Agregar usuario.

- Asignar perfil.
  - Actualizar usuario.
  - Eliminar usuario
- Guardar en el sistema



**Ilustración 29:** Diagrama de caso de uso 11: Administración de portero.

#### 5.1.1.6. DIAGRAMA DE CASO DE USO ESTADISTICAS DEL SISTEMA

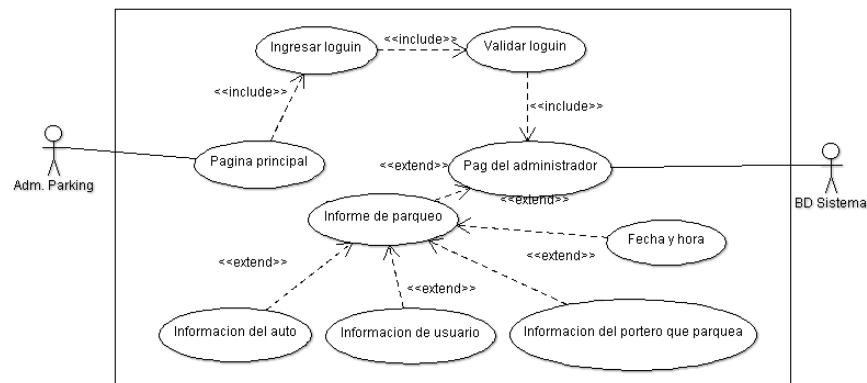
El sistema Easy Parking permite al administrador del sistema tener acceso a un reporte estadístico sobre los parqueos realizados, esto es de mucha utilidad pues se conoce la información precisa de cada paqueo.

Las operaciones que puede realizar el administrador con las estadísticas del sistema son las siguientes:

- Página principal.
- Ingresar loguin
- Validar loguin
- Pag del administrador
- Informe de parqueo



- Información del auto
- Información de usuario
- Información del portero que ingresa el parqueo
- Fecha y hora



**Ilustración 30:** Diagrama de caso de uso 12: Estadísticas del sistema.

### 5.2.2. DOCUMENTACIÓN DE LOS CASOS DE USO.

A continuación se describen los casos de usos más relevantes del sistema Easy Parking. La documentación de los casos de uso está definida de forma muy clara y entendible, identificando a los actores del sistema.

<b>Caso de uso 01</b>	<b>Control de acceso</b>
<b>Objetivos asociados</b>	OBJETIVO 01- Control de acceso
<b>Requisitos asociados</b>	RI-01
<b>Descripción</b>	El sistema Easy Parking deberá permitir la realización de un conjunto de acciones cuando el administrador o el portero intentan ingresar al sistema.
<b>Pre- condición</b>	El administrador o portero deben estar registrados en el sistema de Base de datos de la Easy Parking
<b>Secuencia normal</b>	<b>Pasos y acciones:</b> <ul style="list-style-type: none"> <li>- Ingresar Datos (Username y Password)</li> </ul>

	<ul style="list-style-type: none"> <li>- Validar datos</li> <li>- Conectar con la base de datos</li> <li>- Verificar datos</li> <li>- Ingreso de Adm. Apps Web</li> <li>- Ingreso de administrador del parqueadero</li> <li>- Ingreso del portero del parqueadero</li> <li>- Entran al sistema de Base de datos</li> </ul>
<b>Pos- condición</b>	Si el portero está registrado puede usar los servicios del sistema.
<b>Excepciones</b>	Si alguno de los loguin sale erróneo volver a verificar ingreso del loguin
<b>Rendimiento</b>	
<b>Frecuencia</b>	
<b>Comentarios</b>	Ninguno

**Tabla 13:** Documentación del caso de uso del control de acceso.

<b>Caso de uso 02</b>	<b>Sistema de GPS</b>
<b>Objetivos asociados</b>	OBJETIVO 02- Sistema de GPS.
<b>Requisitos asociados</b>	RI-02
<b>Descripción</b>	El sistema Easy Parking deberá permitir al usuario localizar por medio de GPS su posición y la posición de los parqueaderos
<b>Pre- condición</b>	El usuario de la Apps Web deberá permitir que el sistema conozca su posición por medio del GPS.
<b>Secuencia normal</b>	<b>Pasos y acciones:</b> <ul style="list-style-type: none"> <li>- Ingresar a la Apps Web.</li> <li>- Muestra posición</li> <li>- Busca parqueaderos</li> <li>- Consulta</li> <li>- Ubicación e información</li> <li>- Mostrar resultados</li> </ul>
<b>Pos- condición</b>	Ninguno
<b>Excepciones</b>	Refrescar la página en caso que no salga el mapa o su ubicación.
<b>Rendimiento</b>	
<b>Frecuencia</b>	
<b>Comentarios</b>	Ninguno

**Tabla 14:** Documentación del caso de uso de sistema de GPS.

<b>Caso de uso 03</b>	<b>Administración de parqueaderos.</b>
<b>Objetivos asociados</b>	OBJETIVO 03- Administración de parqueaderos.
<b>Requisitos asociados</b>	RI-03
<b>Descripción</b>	El sistema Easy Parking deberá permitir la administración exitosa de los parqueaderos que maneja el sistema.
<b>Pre- condición</b>	El administrador de la Apps Web debe acceder al sistema sin ningún inconveniente.
<b>Secuencia normal</b>	<b>Pasos y acciones:</b> <ul style="list-style-type: none"> <li>- Ingresar Loguin</li> <li>- Validar Loguin</li> <li>- Escoger ubicación del parqueadero en el mapa <ul style="list-style-type: none"> <li>• Agregar parqueaderos</li> <li>• Modificar Parqueaderos</li> <li>• Eliminar Parqueaderos</li> </ul> </li> <li>- Guardar configuración</li> </ul>
<b>Pos- condición</b>	Si el administrador de la Apps Web esta logueado correctamente puede hacer las configuraciones pertinentes del sistema.
<b>Excepciones</b>	Si el administrador no ha iniciado sesión volver a verificar el ingreso del loguin
<b>Rendimiento</b>	
<b>Frecuencia</b>	
<b>Comentarios</b>	Ninguno

**Tabla 15:** Documentación del caso de uso de administración de parqueaderos.

<b>Caso de uso 04</b>	<b>Navegación.</b>
<b>Objetivos asociados</b>	OBJETIVO 04- Navegación.
<b>Requisitos asociados</b>	RI-04
<b>Descripción</b>	El sistema Easy Parking deberá permitir la búsqueda de diferentes parqueaderos con relación a los parámetros establecidos en el sistema.
<b>Pre- condición</b>	El usuario debe ingresar a la Apps Web.
<b>Secuencia normal</b>	<b>Pasos y acciones:</b> <ul style="list-style-type: none"> <li>- Explorar menú principal</li> <li>- Mapa</li> <li>- Parqueaderos</li> </ul>

	<ul style="list-style-type: none"> <li>- Información de parqueaderos</li> <li>- Loguin</li> <li>- Página de información según rol</li> </ul>
<b>Pos- condición</b>	Si el usuario accedió al sistema exitosamente puede encontrar las opciones de navegación según el rol que cumple en la Apps Web.
<b>Excepciones</b>	Ninguno.
<b>Rendimiento</b>	
<b>Frecuencia</b>	
<b>Comentarios</b>	Ninguno

**Tabla 16:** Documentación del caso de uso de Navegación.

<b>Caso de uso 05</b>	<b>Búsqueda.</b>
<b>Objetivos asociados</b>	OBJETIVO 05- Búsqueda.
<b>Requisitos asociados</b>	RI-05
<b>Descripción</b>	El sistema Easy Parking deberá permitir la búsqueda de diferentes parqueaderos con relación a los parámetros establecidos en el sistema.
<b>Pre- condición</b>	El usuario debe ingresar a la Apps Web.
<b>Secuencia normal</b>	<b>Pasos y acciones:</b> <ul style="list-style-type: none"> <li>- Mapa</li> <li>- Ubicación actual</li> <li>- Buscar parqueadero</li> <li>- Mostrar Parqueadero</li> <li>- Ver información Parqueadero</li> </ul>
<b>Pos- condición</b>	Si el usuario accedió al sistema exitosamente puede encontrar las opciones de búsquedas de parqueaderos.
<b>Excepciones</b>	Ninguno.
<b>Rendimiento</b>	
<b>Frecuencia</b>	
<b>Comentarios</b>	Ninguno

**Tabla 17:** Documentación del caso de uso de búsqueda.

<b>Caso de uso 07</b>	<b>Administración de portero.</b>
<b>Objetivos asociados</b>	OBJETIVO 07- Administración de portero.
<b>Requisitos asociados</b>	RI-07

<b>Descripción</b>	El sistema Easy Parking le permitirá al administrador la posibilidad de agregar, modificar y eliminar los porteros que se harán cargo del parqueadero.
<b>Pre- condición</b>	El administrador debe loguiarse para poder acceder al sistema Easy Parking y cumplir con la función de administrar portero.
<b>Secuencia normal</b>	<b>Pasos y acciones:</b> <ul style="list-style-type: none"> <li>- Pagina principal</li> <li>- Ingresar loguin.</li> <li>- Validar loguin.</li> <li>- Página del administrador</li> <li>- Información del parqueadero</li> <li>- Información parqueo</li> <li>- Administrar portero.</li> <li>- Agregar usuario.</li> <li>- Asignar perfil.</li> <li>- Actualizar usuario.</li> <li>- Eliminar usuario</li> <li>- Guardar en el sistema</li> </ul>
<b>Pos- condición</b>	Si el administrador esta logueado exitosamente puede estar en la capacidad de agregar, modificar y eliminar porteros.
<b>Excepciones</b>	Si el administrador no ha iniciado sesión volver a verificar loguin.
<b>Rendimiento</b>	
<b>Frecuencia</b>	
<b>Comentarios</b>	Ninguno

**Tabla 18:** Documentación del caso de Administración de portero.

<b>Caso de uso 08</b>	<b>Estadísticas del sistema</b>
<b>Objetivos asociados</b>	OBJETIVO 08- Estadísticas del sistema
<b>Requisitos asociados</b>	RI-08
<b>Descripción</b>	El sistema Easy Parking le permitirá al administrador visualizar un reporte estadístico del sistema, en el cual se encontrara la información de los usuarios que accedieron al servicio con sus respectivos vehículos y algunos datos de ellos además de que portero registro el parqueo, la fecha y la hora de dicho parqueo.
<b>Pre- condición</b>	El administrador debe loguiarse para poder acceder la información que se encuentra en el sistema Easy Parking.

<b>Secuencia normal</b>	<b>Pasos y acciones:</b> <ul style="list-style-type: none"> <li>- Página principal.</li> <li>- Ingresar login</li> <li>- Validar login</li> <li>- Pag del administrador</li> <li>- Informe de parqueo <ul style="list-style-type: none"> <li>• Información del auto</li> <li>• Información de usuario</li> <li>• Información del portero que ingresa el parqueo</li> <li>• Fecha y hora</li> </ul> </li> </ul>
<b>Pos- condición</b>	Si el administrador esta logueado exitosamente puede estar en la capacidad de ver la información del sistema
<b>Excepciones</b>	Si el administrador no ha iniciado sesión volver a verificar login.
<b>Rendimiento</b>	
<b>Frecuencia</b>	
<b>Comentarios</b>	Ninguno

**Tabla 19:** Documentación del caso de Estadísticas del sistema.

## 5.2. REQUISITOS NO FUNCIONALES

Un requisito no Funcional o atributo de Calidad hacen relación a las características del sistema, es decir, es un requisito que especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, ya que éstos corresponden a los requisitos funcionales. Por tanto, se refieren a todos los requisitos que no describen información a guardar, ni funciones a realizar.

Estos requerimientos son adicionales a los requerimientos funcionales que debe cumplir el sistema y corresponde a aspectos tales como: La disponibilidad, flexibilidad, ayuda del sistema, entre otras, los cuales se describen a continuación:

Los requisitos no funcionales del sistema Easy Parking son los siguientes:

### **5.2.1. AYUDA DEL SISTEMA**

El sistema debe ser accesible y adaptable a las necesidades de cada usuario, por lo tanto proporciona una documentación que garantiza la correcta manipulación del sistema.

### **5.2.2. CONTROL DE PROCESOS**

El sistema no debe permitir el cierre de una operación hasta que todos sus procesos, subprocesos y tareas relacionados, hayan sido terminados y cerrados satisfactoriamente.

### **5.2.3. INTEGRIDAD DEL SISTEMA EASY PARKING**

El sistema Easy Parking deberá asegurar la integridad de los parqueaderos que se administran garantizando que el sistema no tiende a un estado de entropía.

### **5.2.4. DISPONIBILIDAD DEL SISTEMA EASY PARKING**

Al acceso al sistema Easy Parking deberá estar disponible en las horas de atención en los parqueaderos para los usuarios que deseen acceder al servicio.

## **5.3. DIAGRAMA DE SECUENCIA**

Un diagrama de secuencia muestra una interacción de un conjunto de objetos en una aplicación que representa la secuencia de mensajes entre las instancias de clases, componentes, subsistemas o actores, es decir, El diagrama de secuencias en UML muestra la forma en que los objetos se comunican entre sí al transcurrir el tiempo. El tiempo fluye hacia abajo en el diagrama y muestra el flujo de control de un participante a otro. El diagrama de secuencia contiene detalles de la implementación del escenario, incluyéndolo los objetos y clases que se usan para implementar el escenario y mensajes intercambiados entre los objetos

En un diagrama de secuencia se indicarán los módulos o clases que forman parte del programa y las llamadas que se hacen en cada uno de ellos para realizar una tarea determinada. Se realizan diagramas de secuencia para definir acciones que se pueden realizar en la aplicación en cuestión. Así, en el caso de una aplicación para jugar al ajedrez, se podrían realizar diagramas de secuencia para “jugar una partida” o bien para acciones más específicas como “mover pieza”.

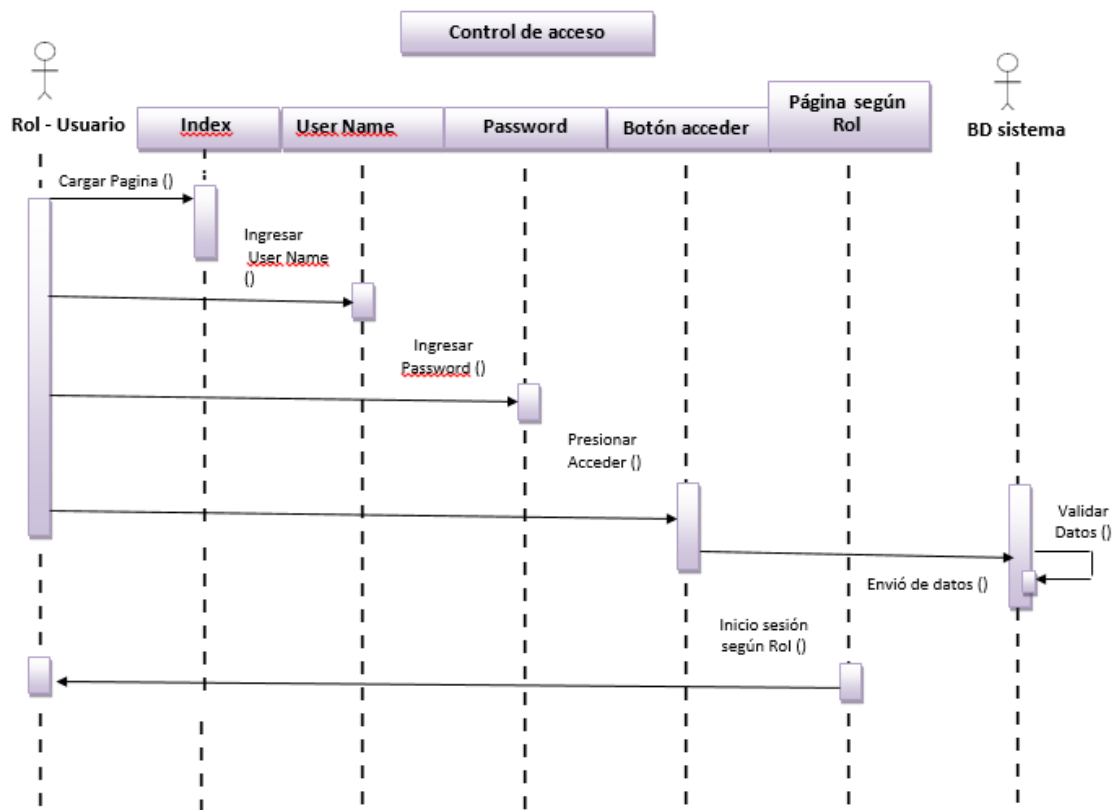
Existen dos tipos de mensajes: sincrónicos y asincrónicos. Los mensajes sincrónicos corresponden con llamadas a métodos de objetos que recibe el mensaje. El objeto que envía el mensaje queda bloqueado hasta que termine la llamada. Este tipo de mensaje se representa con flechas con la cabeza llena. Los mensajes asincrónicos terminan inmediatamente y crean un nuevo hilo de ejecución dentro de la secuencia se representan con flechas con la cabeza abierta.

Ahora mostraremos los diagramas de involucran el sistema Easy Parking en sus diferentes módulos:

#### **5.3.1. DIAGRAMA DE SECUENCIA PARA EL CONTROL DE ACCESO**

En este diagrama se describen los pasos para que el actor según su rol ingrese al sistema Easy Parking a realizar sus labores administrativas referentes al sistema, como tal no se describirá el acceso a los usuarios normales por que estos no necesitaran estar registrados para acceder al sistema.

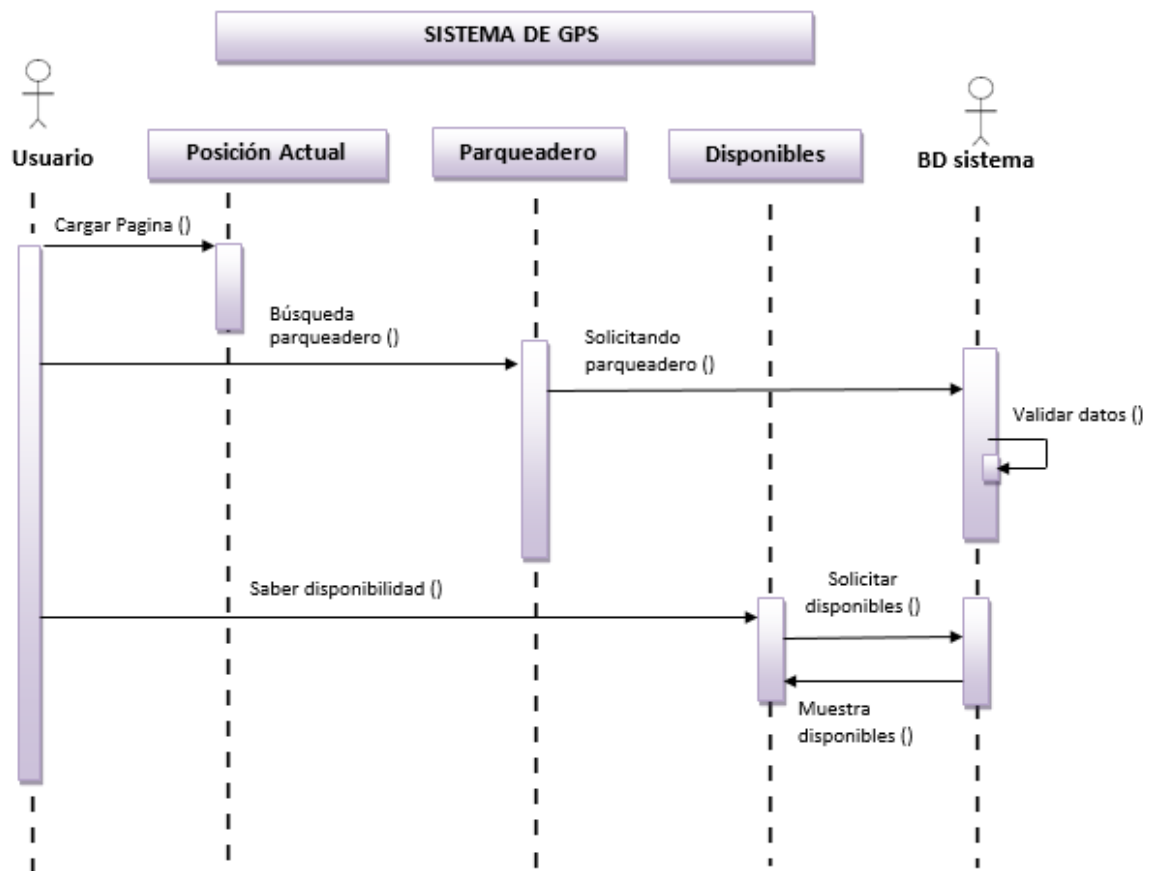




**Ilustración 31:** Diagrama de Secuencia 01 Control de acceso.

### 5.3.2. DIAGRAMA DE SECUENCIA PARA EL SISTEMA DE GPS

En este diagrama se describe como el usuario del sistema Easy Parking por medio del sistema de GPS localiza los parqueaderos que hacen parte de la Apps Web.

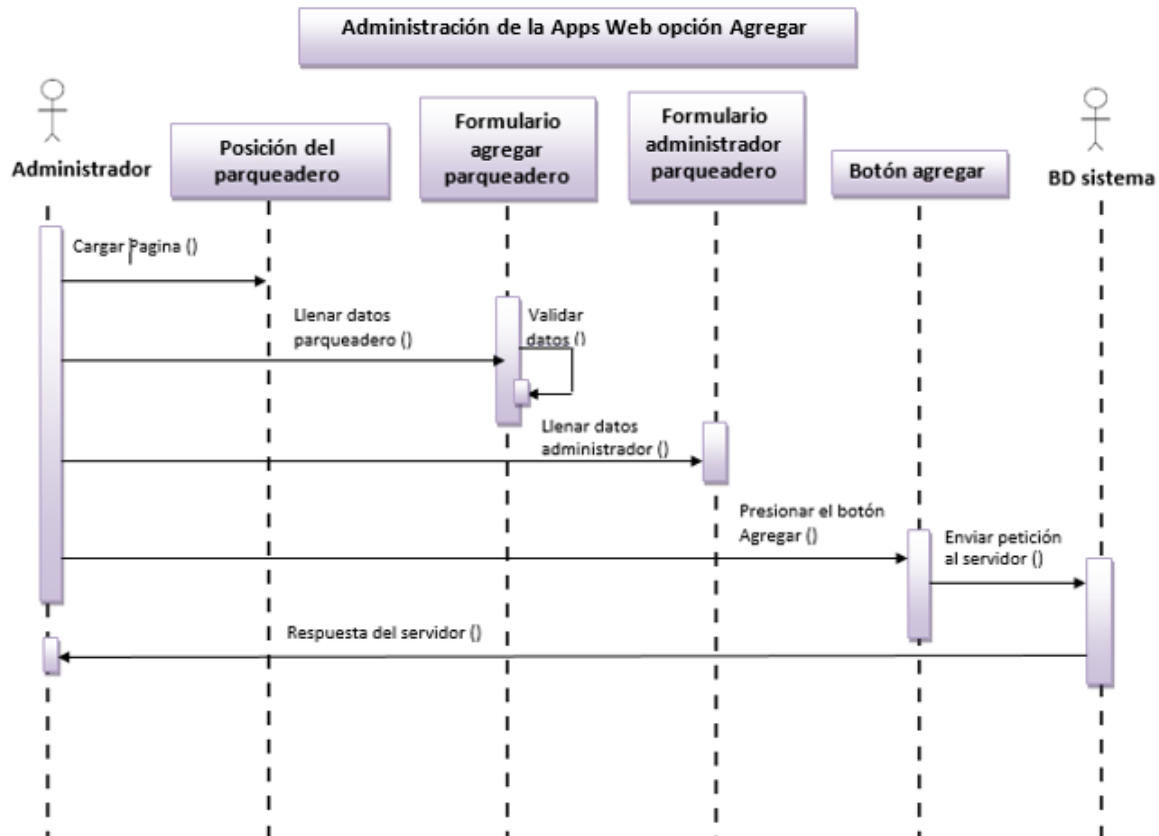


**Ilustración 32:** Diagrama de Secuencia 02 Sistema de GPS.

### 5.3.3. DIAGRAMA DE SECUENCIA PARA LA ADMINISTRACION DE PARQUEADEROS

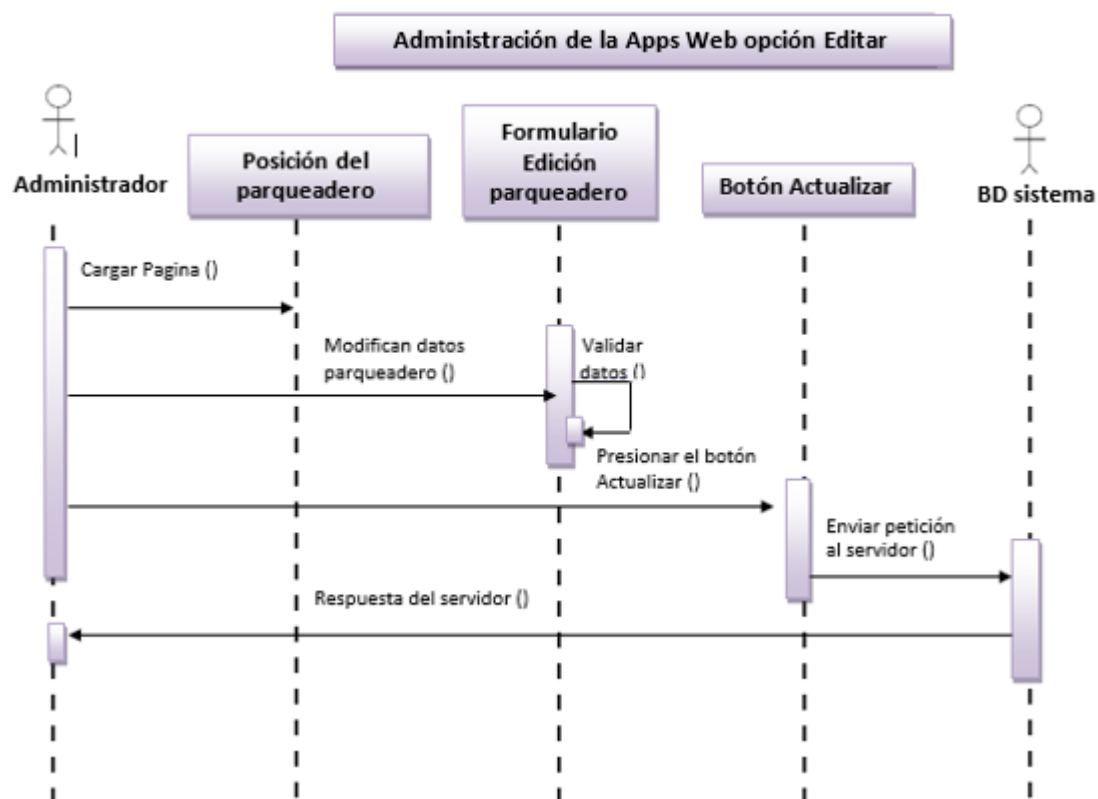
En los siguientes diagramas se describirá los pasos necesarios que el administrador de la Apps Web realiza para cumplir con labores administrativas como la de agregar, actualizar y eliminar la gama de parqueaderos que existen en el sistema Easy Parking.

- **Administración de parqueaderos opción agregar:** En este diagrama se describe los pasos que realiza el administrador de la Apps Web para acceder al sistema a realizar el ingreso de parqueaderos.



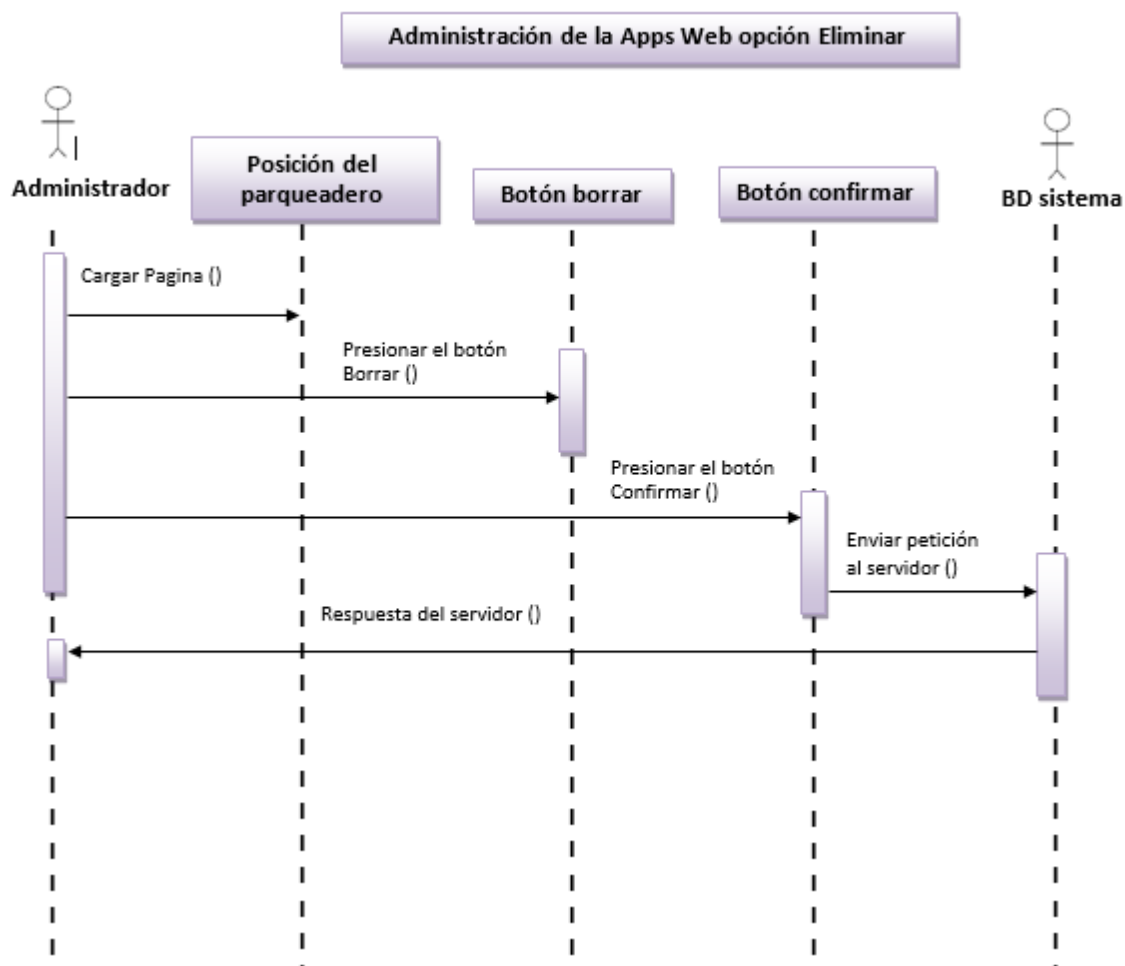
**Ilustración 33:** Diagrama de Secuencia 03 Administración de parqueaderos opción agregar.

- **Administración de parqueaderos opción Modifica:** En este diagrama se describe los pasos que realiza el administrador de la Apps Web para acceder al sistema a realizar la modificación de los parqueaderos que requieran ser modificados.



**Ilustración 34:** Diagrama de Secuencia 03 Administración de productos opción modificar.

- **Administración de parqueaderos opción eliminar:** En este diagrama se describe los pasos que realiza el administrador de la Apps Web para acceder al sistema a realizar la eliminación de los parqueaderos que lo requieran.



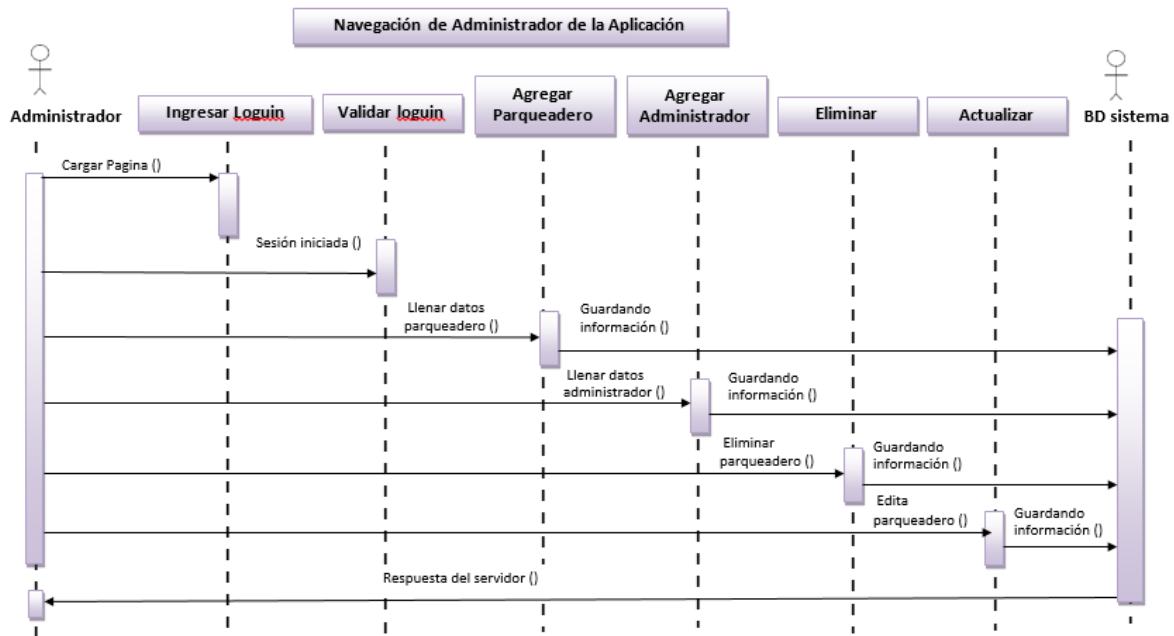
**Ilustración 35:** Diagrama de Secuencia 03 Administración de parqueaderos opción eliminar.

#### 5.3.4. DIAGRAMA DE SECUENCIA NAVEGACIÓN

La Apps Web ofrece a todos los actores una fácil navegación a través de sus distintos módulos. Los siguientes diagramas de secuencia describen como se enlazan e interactúan los distintos eventos de la aplicación con los diferentes actores que tiene el sistema.

##### ➤ **DIAGRAMA DE SECUENCIA NAVEGACION DEL ADMINISTRADOR DE LA APPS WEB**

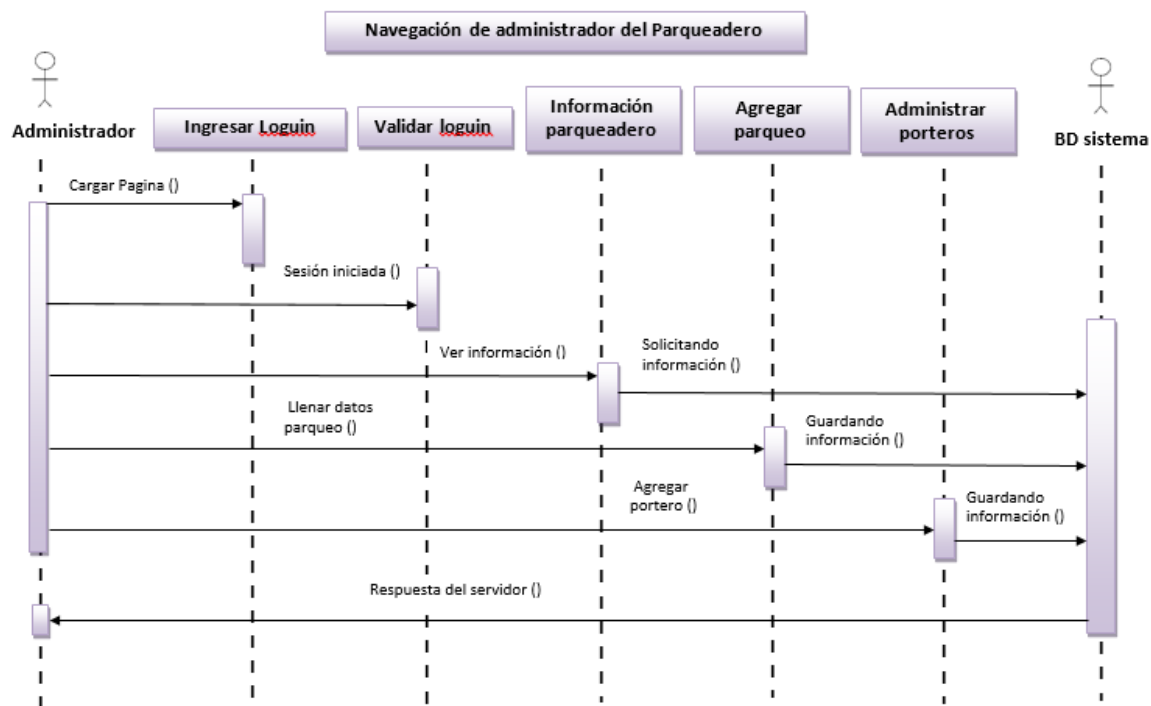
Este diagrama de secuencia describe los pasos que realiza el administrador de la Apps Web para acceder al sistema y cumplir a cabalidad con sus funciones.



**Ilustración 36:** Diagrama de Secuencia 04 Navegación del Administrador de la Apps Web.

### ➤ DIAGRAMA DE SECUENCIA NAVEGACION DEL ADMINISTRADOR DEL PARQUEADERO

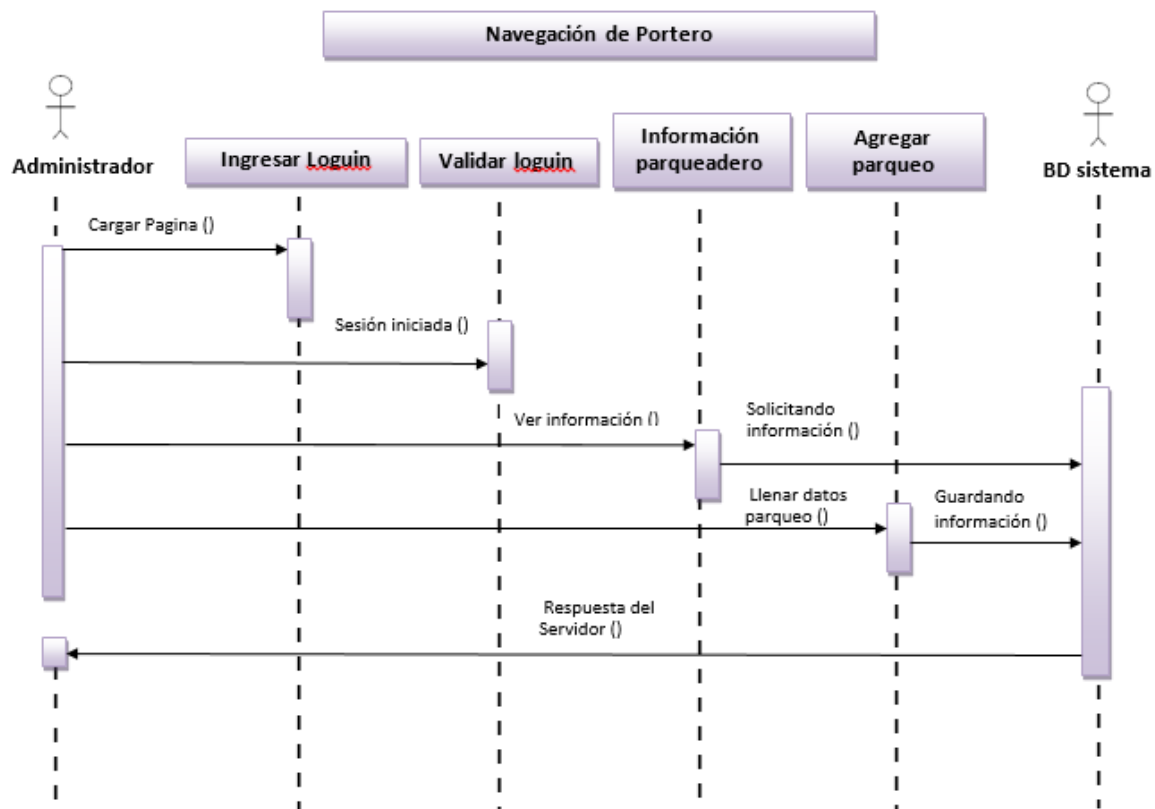
Este diagrama de secuencia describe los pasos que realiza el administrador del parqueadero para acceder al sistema y cumplir a cabalidad con sus funciones.



**Ilustración 37:** Diagrama de Secuencia 04 Navegación del Administrador del parqueadero.

### ➤ DIAGRAMA DE SECUENCIA NAVEGACION DE PORTERO

Este diagrama de secuencia describe los pasos que realiza portero del parqueadero para acceder al sistema y cumplir a cabalidad con sus funciones.

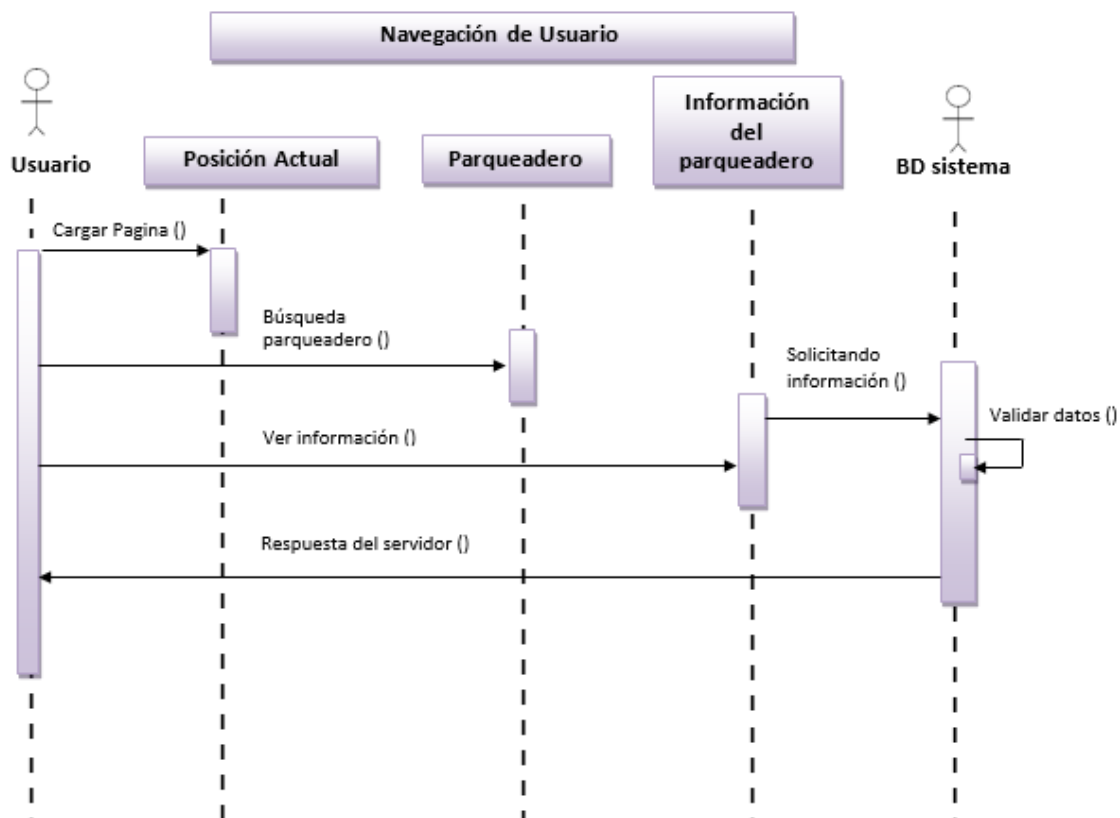


**Ilustración 38:** Diagrama de Secuencia 04 Navegación de portero.

### ➤ DIAGRAMA DE SECUENCIA NAVEGACION DE USUARIO

Este diagrama de secuencia describe los pasos que realiza el usuario para acceder al mapa y conocer su funcionalidad e información sobre los parqueaderos y su disponibilidad.

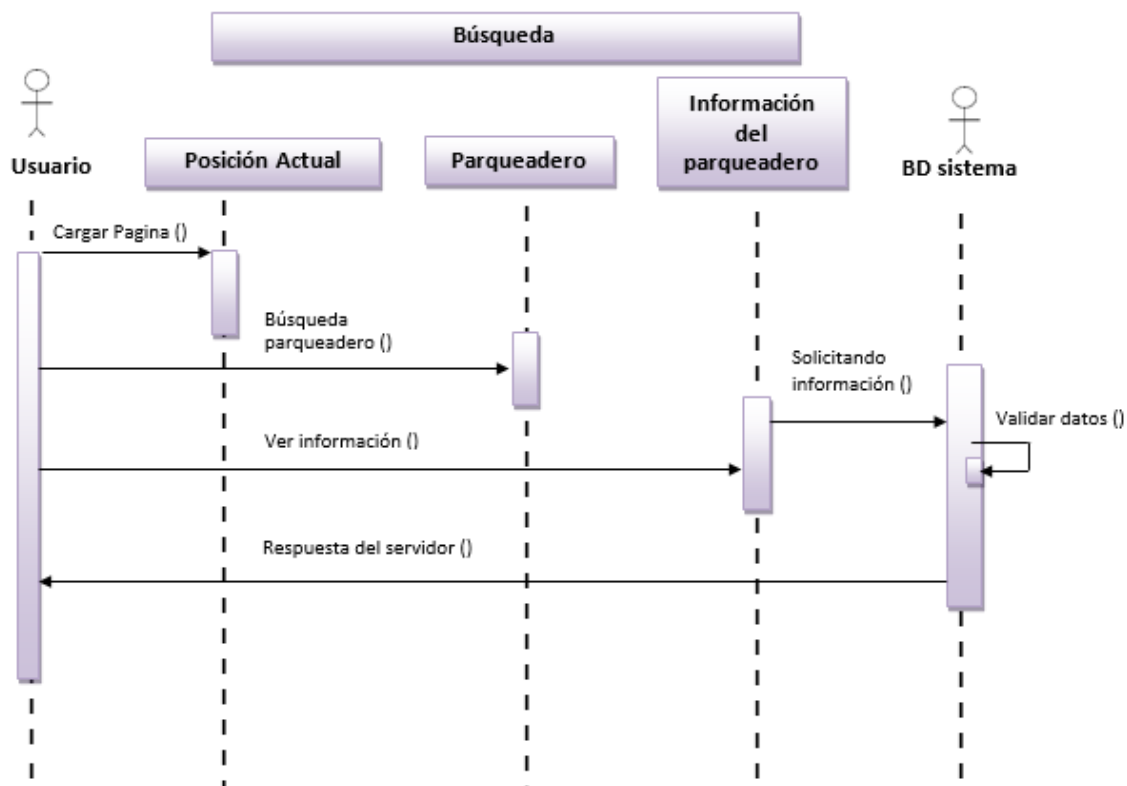




**Ilustración 39:** Diagrama de Secuencia 04 Navegación de Usuario.

### 5.3.5. DIAGRAMA DE SECUENCIA DE BÚSQUEDA

En este diagrama se describe como el usuario ingresa al sistema y eligen la opción de búsqueda que tiene éste para la obtención de parqueaderos.

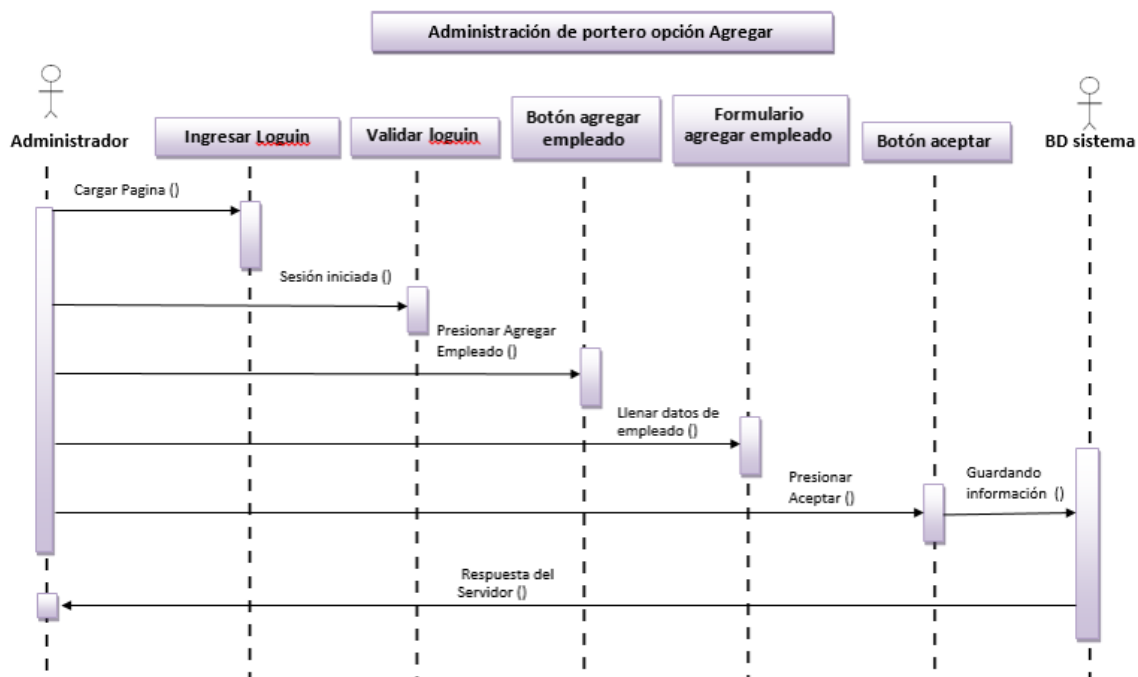


**Ilustración 40:** Diagrama de Secuencia 05 Búsqueda.

### 5.3.6. DIAGRAMA DE SECUENCIA DE ADMINISTRACION DE PORTERO

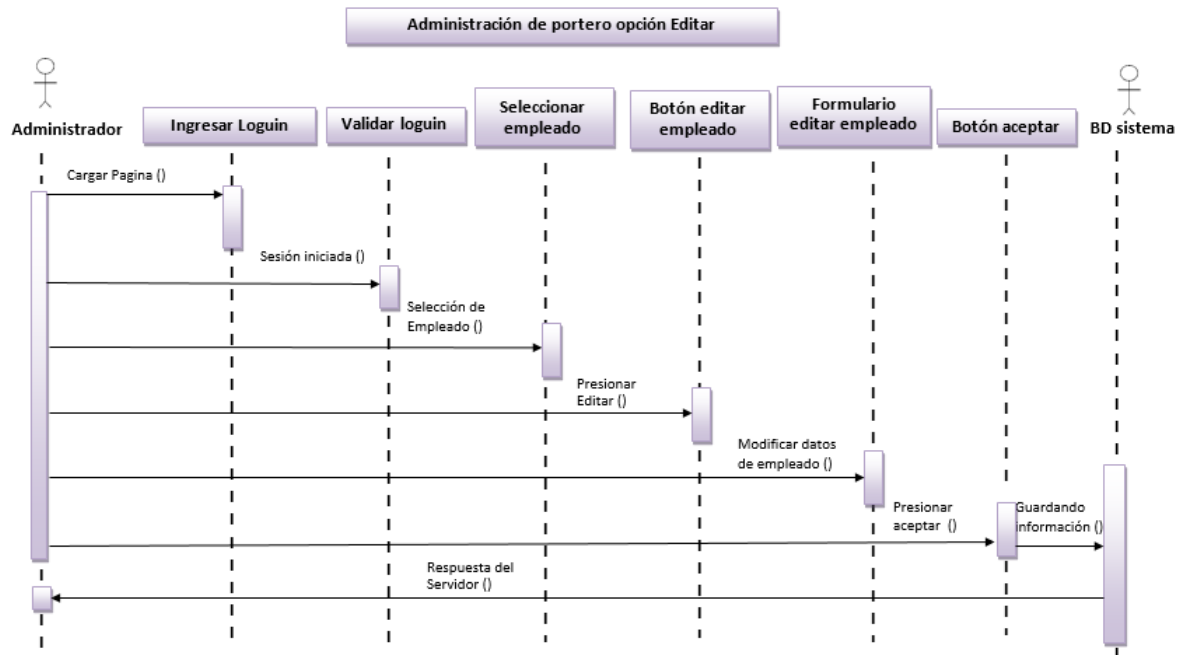
En este diagrama se describe los pasos que realiza el administrador del sistema para registrar a los porteros del parqueadero, el administrador puede agregar actualizar y eliminar.

- **Administración de porteros opción agregar:** En este diagrama se describe los pasos que realiza el administrador del parqueadero para acceder al sistema a realizar el ingreso de un nuevo portero



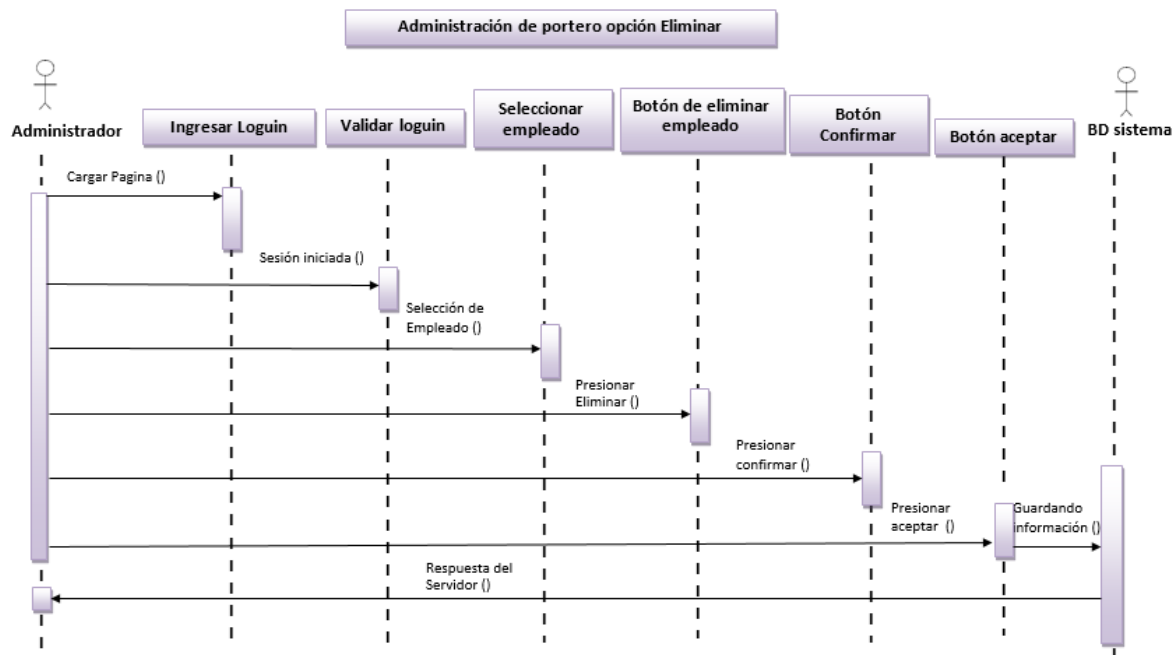
**Ilustración 41:** Diagrama de Secuencia 07 Administración de portero opción agregar.

- **Administración de portero opción Modifica:** En este diagrama se describe los pasos que realiza el administrador del parqueadero para acceder al sistema a realizar la modificación de los porteros que requieran ser modificados.



**Ilustración 42:** Diagrama de Secuencia 07 Administración de porteros opción modificar.

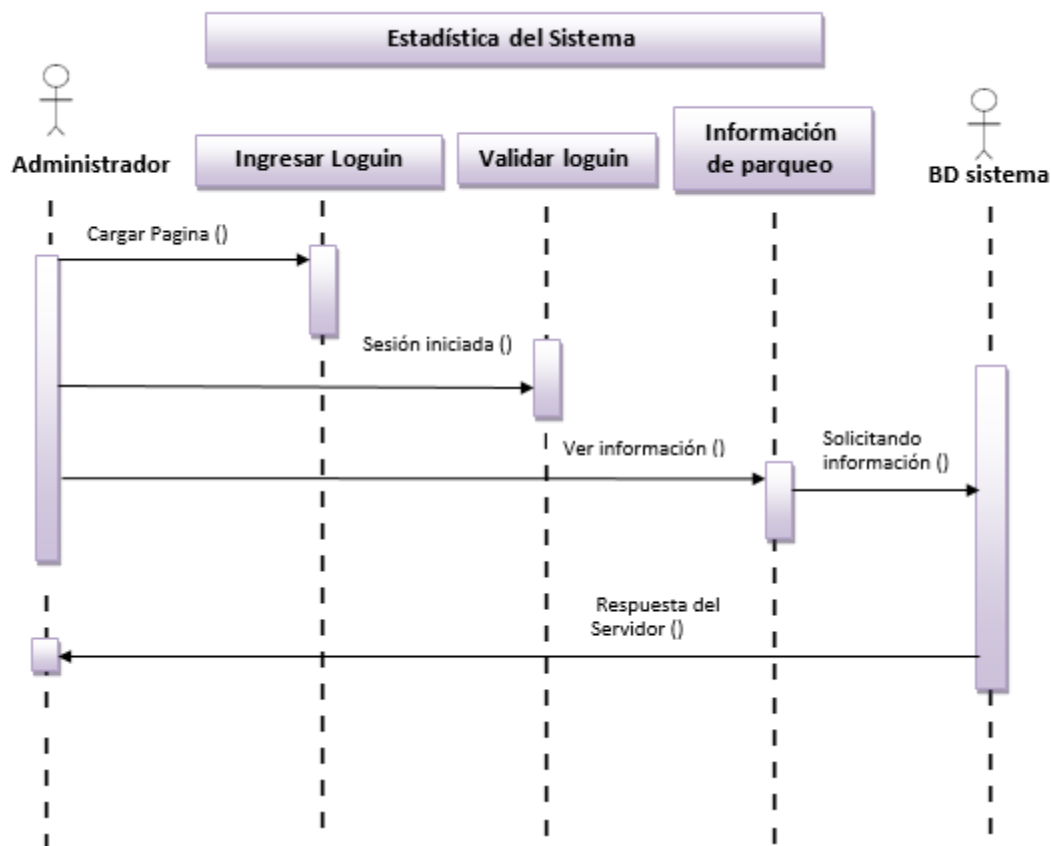
- **Administración de porteros opción eliminar:** En este diagrama se describe los pasos que realiza el administrador del parqueadero para acceder al sistema a realizar la eliminación de los porteros que lo requieran.



**Ilustración 43:** Diagrama de Secuencia 07 Administración de portero opción eliminar.

### 5.3.7. DIAGRAMA DE SECUENCIA ESTADISTICAS DEL SISTEMA

En este diagrama se describe los pasos que realiza el administrador del sistema para conocer un informe con los datos más relevantes de la aplicación, es decir, los datos del usuario, del vehículo parqueado, del personal que hace el ingreso la fecha y la hora del parqueo.



**Ilustración 44:** Diagrama de Secuencia 08 Estadísticas del sistema.

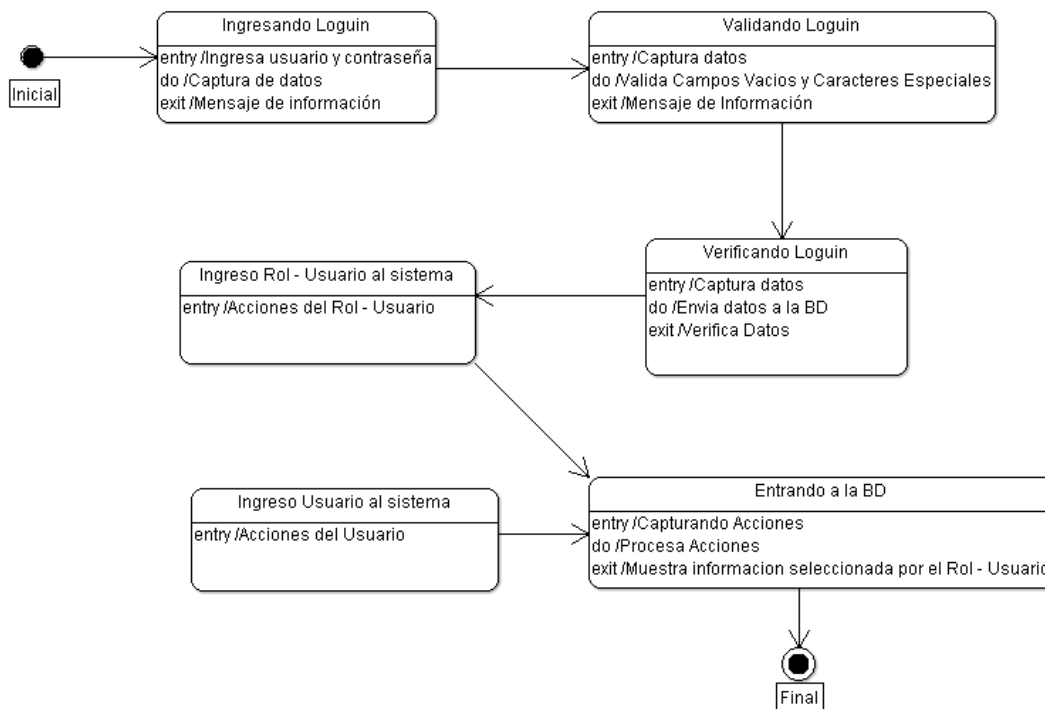
### 5.3.DIAGRAMAS DE ESTADO

En esta parte utilizaremos diagramas de estados que muestran una máquina de estado, son útiles para modelar la vida de un objeto. Un diagrama de estados muestra el flujo de control entre estados (en qué estados posibles puede estar “cierto algo” y como se producen los cambios entre dichos estados) y que utilizamos para identificar cada una de las rutas o caminos que puede tomar un flujo de información luego de ejecutarse cada proceso. Permite identificar bajo qué argumentos se ejecuta cada uno de los procesos y en qué momento podrían tener una variación.

En otras palabras los diagramas de estados permiten visualizar de una forma secuencial la ejecución de cada uno de los procesos donde actúan los actores del sistema Easy Parking

### 5.3.1. DIAGRAMA DE ESTADO PARA EL CONTROL DE ACCESO

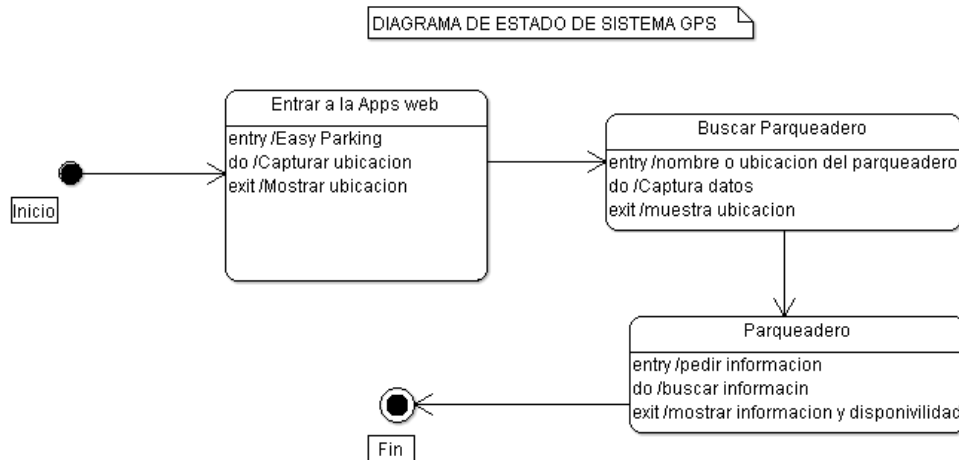
En el siguiente diagrama se describen las acciones de los usuarios del sistema según su rol cuando realizan la entrada a la Apps Web.



**Ilustración 45:** Diagrama de Estado 01 Control de acceso.

### 5.3.8. DIAGRAMA DE ESTADO PARA EL SISTEMA DE GPS

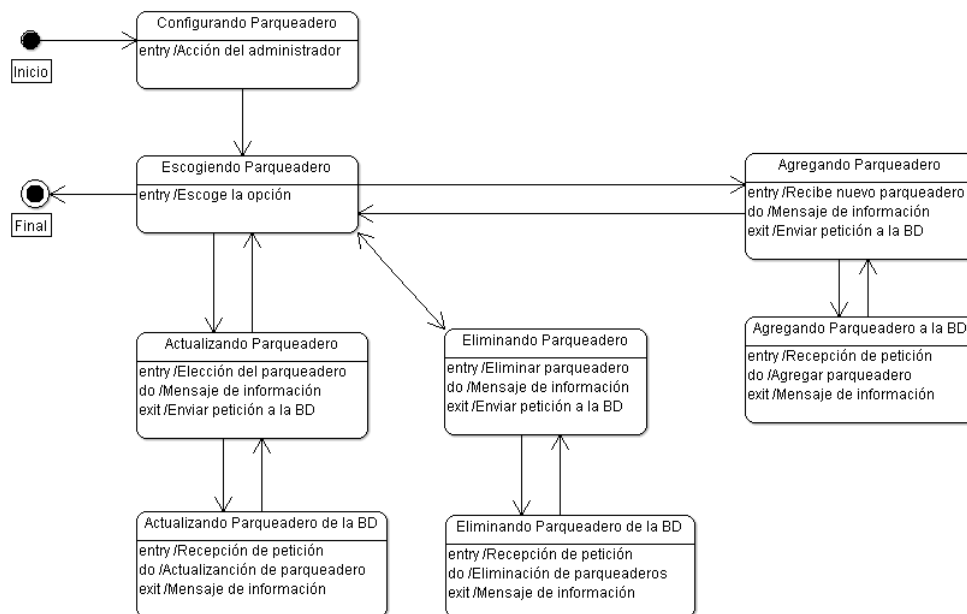
En el siguiente diagrama se describen las acciones del usuario cuando accede a la Apps Web y el sistema GPS muestra la ubicación de los parqueaderos.



**Ilustración 46:** Diagrama de Estado 02 Sistema de GPS.

### 5.3.9. DIAGRAMA DE ESTADO PARA LA ADMINISTRACION DE PARQUEADEROS

El siguiente diagrama nos muestra las acciones del administrador de la Apps Web cuando configura los parqueaderos.



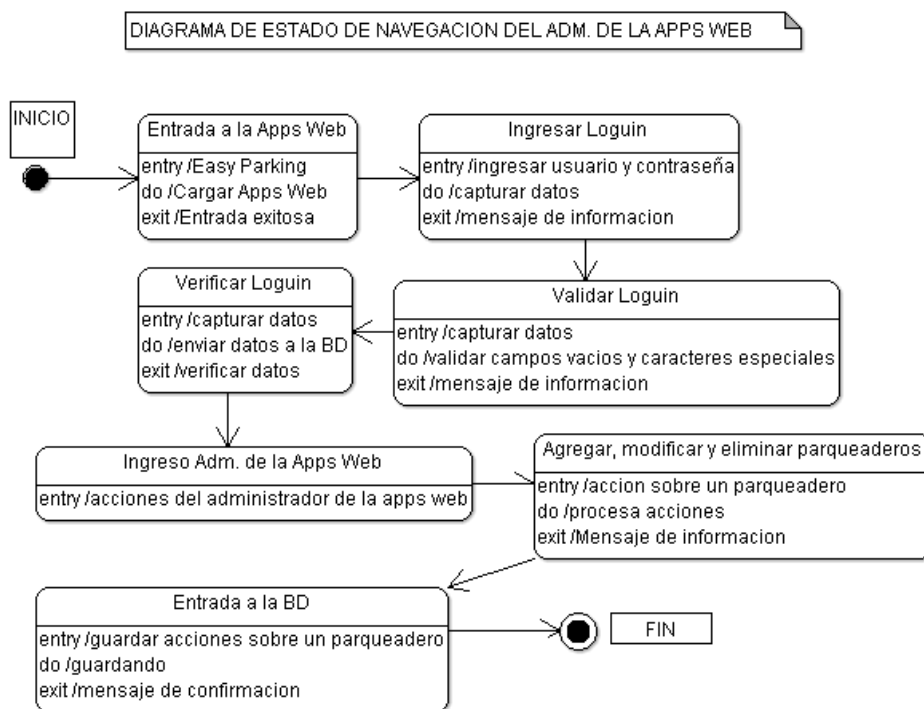
**Ilustración 47:** Diagrama de Estado 03 Administración de Parqueaderos.



### 5.3.10. DIAGRAMA DE ESTADO DE NAVEGACION

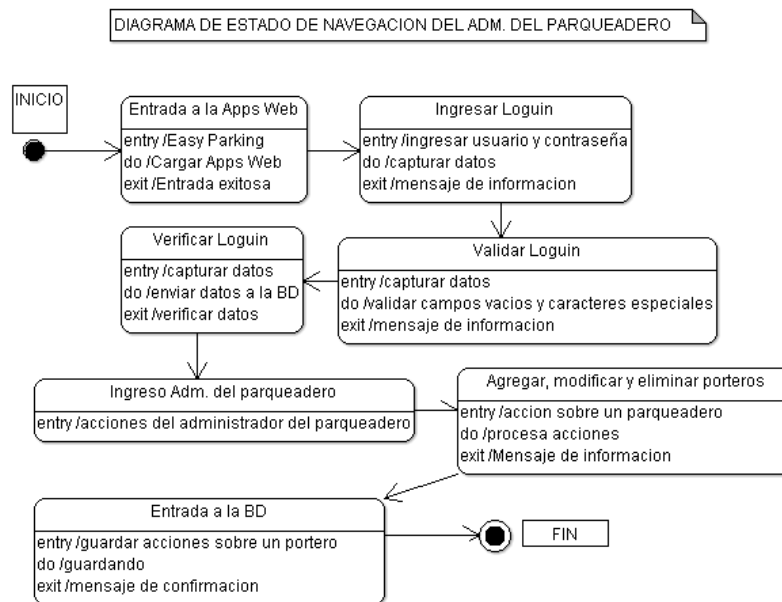
En los siguientes diagramas se describen las acciones del cada actor cuando accede a la Apps Web y las diferentes navegaciones que tiene cada uno según su rol.

#### ➤ DIAGRAMA DE ESTADO DE NAVEGACION DEL ADMINISTRADOR DE LA APPS WEB



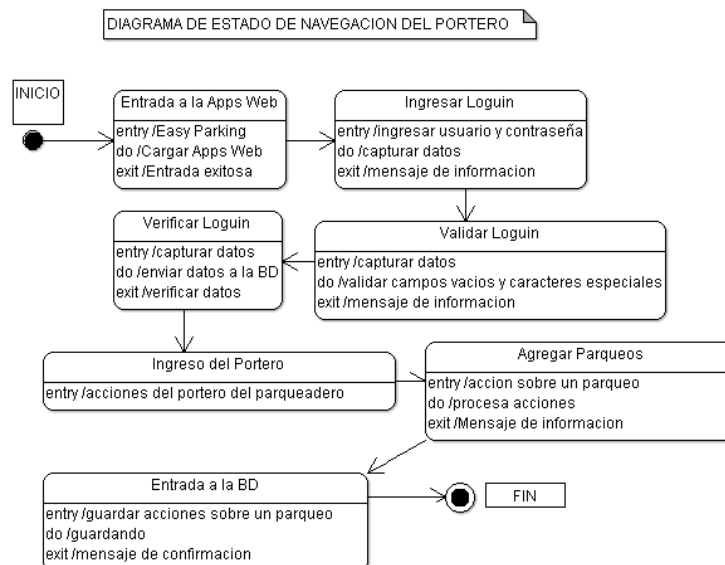
**Ilustración 48:** Diagrama de Estado 04 Navegación del Adm. De la Apps web.

➤ **DIAGRAMA DE ESTADO DE NAVEGACION DEL ADMINISTRADOR DEL PARQUEADERO**



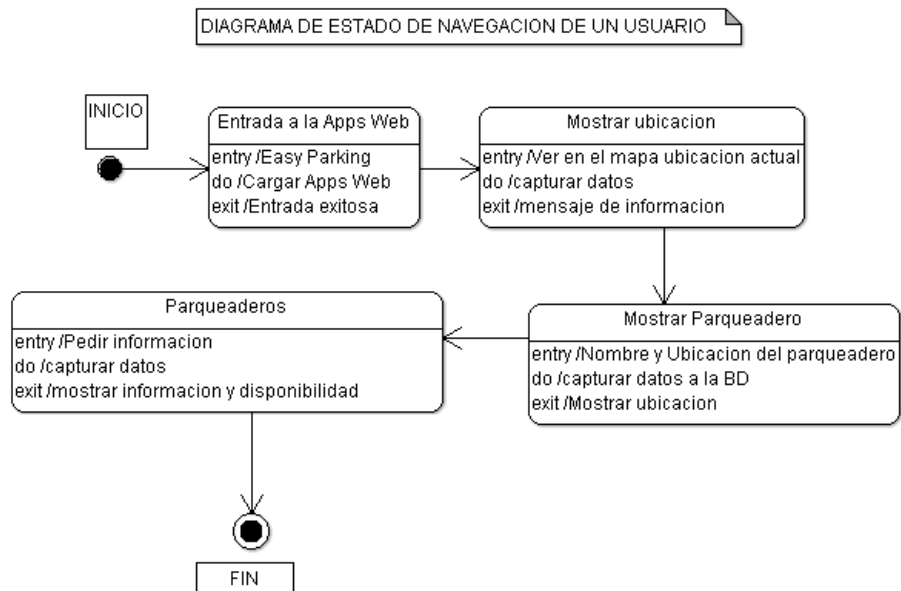
**Ilustración 49:** Diagrama de Estado 04 Navegación del Admon. Del parqueadero.

➤ **DIAGRAMA DE ESTADO DE NAVEGACION DEL PORTERO DEL PARQUEADERO**



**Ilustración 50:** Diagrama de Estado 04 Navegación del Portero Del parqueadero.

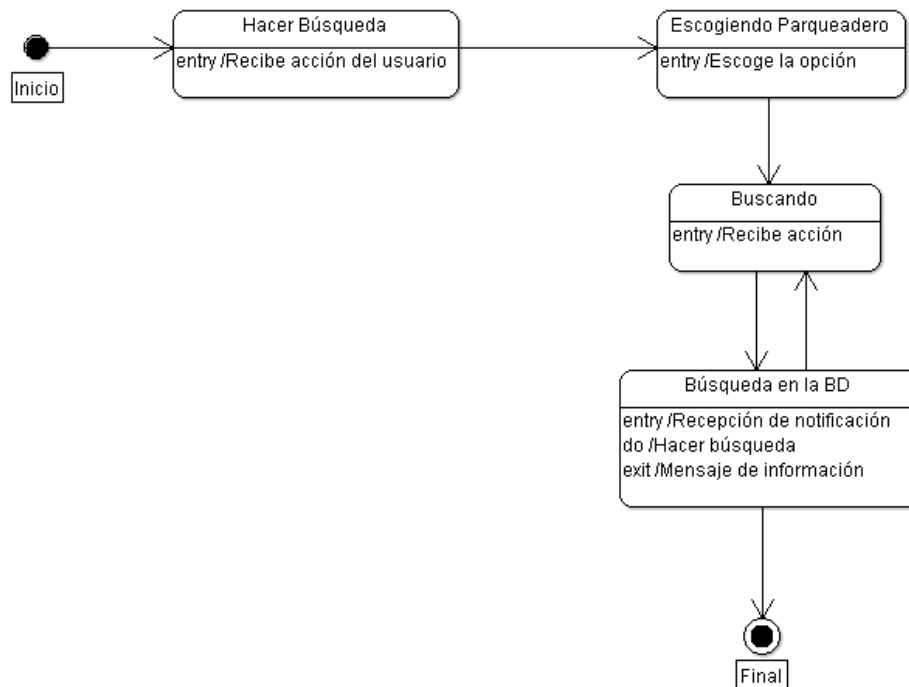
## ➤ DIAGRAMA DE ESTADO DE NAVEGACION DEL USUARIO



**Ilustración 51:** Diagrama de Estado 04 Navegación del Usuario

### 5.3.11. DIAGRAMA DE ESTADO PARA BÚSQUEDA

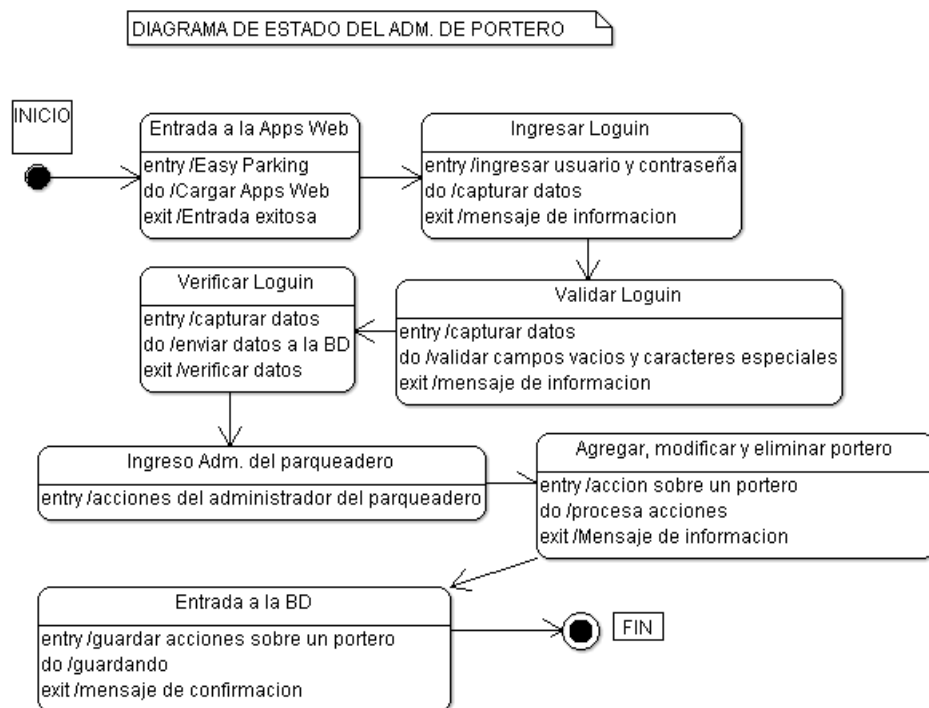
En el siguiente diagrama se describe las acciones que realiza un usuario para hacer sus búsquedas, para así lograr su objetivo que es conseguir sus parqueaderos lo más cercano posible a su destino.



**Ilustración 52:** Diagrama de Estado 05 Búsqueda.

### 5.3.12. DIAGRAMA DE ESTADO PARA LA ADMINISTRACIÓN DE PORTEROS

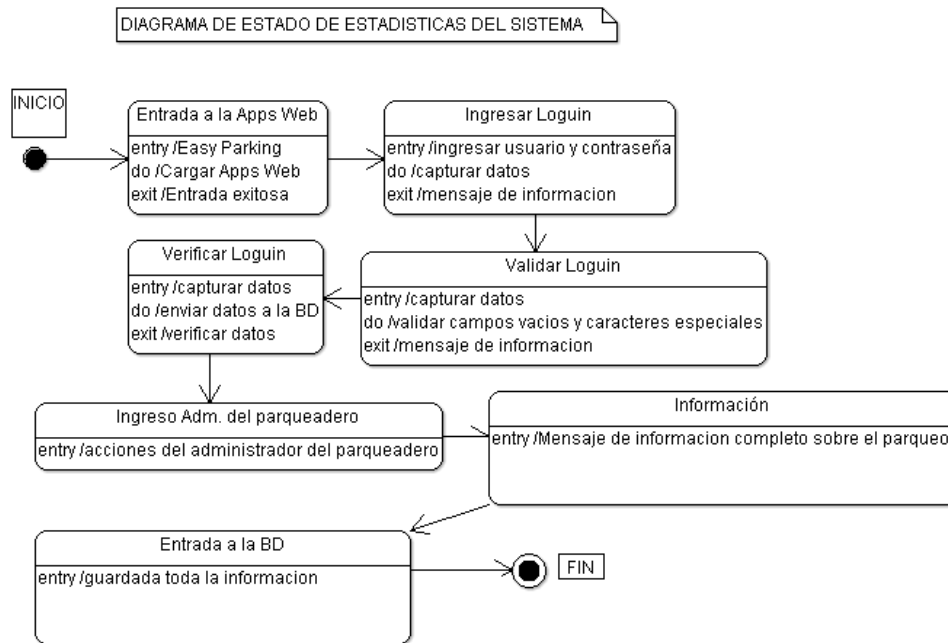
En el siguiente diagrama se describen las acciones del administrador del parqueadero cuando hace las operaciones pertinentes para registrar a los porteros que van a interactuar con el sistema y asignar los parqueos.



**Ilustración 53:** Diagrama de Estado 07 Administración de Portero

### 5.3.13. DIAGRAMA DE ESTADO PARA ESTADISTICAS DEL SISTEMA

En el siguiente diagrama se describen la información que observa el administrador del parqueadero cuando ingresa al sistema, esta información contiene datos sobre el usuario que necesito del servicio, datos del vehículo que ingreso, que portero estaba de turno en ese instante y la fecha y la hora.



**Ilustración 54:** Diagrama de Estado 08 Estadísticas del sistema

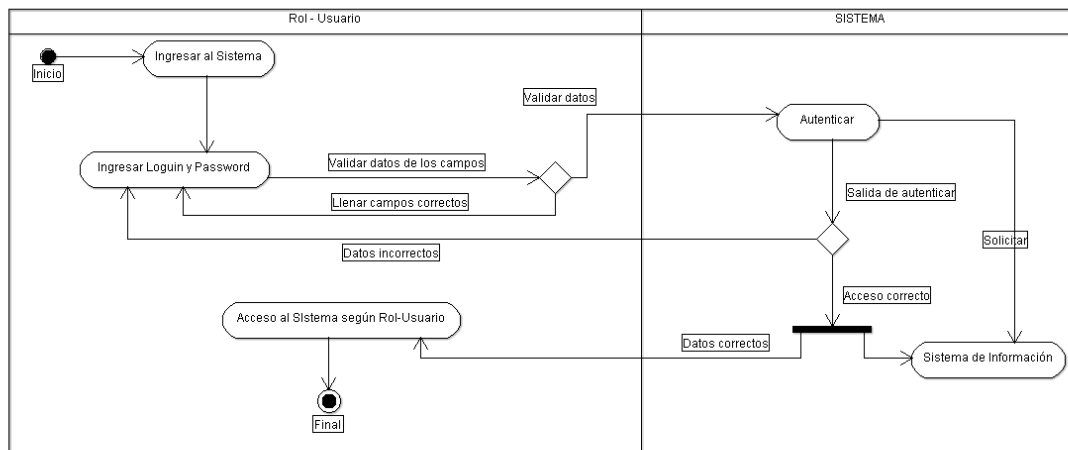
#### 5.4.DIAGRAMA DE ACTIVIDAD

En el lenguaje de Modelado Unificado, un diagrama de actividades se utiliza para mostrar una visión simplificada de lo que ocurre durante una operación o proceso. Un diagrama de actividades representa los flujos de trabajo paso a paso (Transición de una actividad a otra) de negocio y operacionales de los componentes en un sistema. Un diagrama de Actividades muestra el flujo de control general.

Los siguientes son los diagramas que involucra el sistema de administración de préstamos de material bibliográfico de Tienda Virtual.

#### 5.4.1. DIAGRAMA DE ACTIVIDAD DE CONTROL DE ACCESO

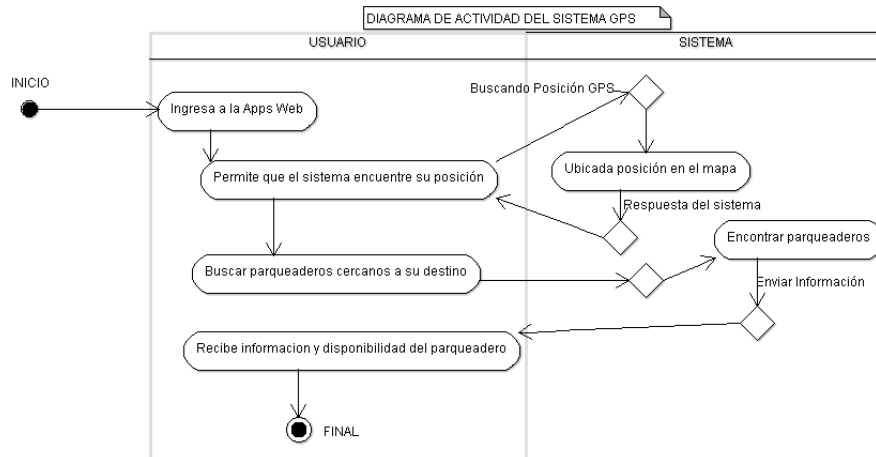
En el siguiente diagrama se muestran las actividades que realiza el administrador para ingresar al sistema como llenar formulario donde se verifica la validez de los datos.



**Ilustración 55:** Diagrama de Actividad 01 Control de acceso.

#### 5.4.2. DIAGRAMA DE ACTIVIDAD DE SISTEMA GPS

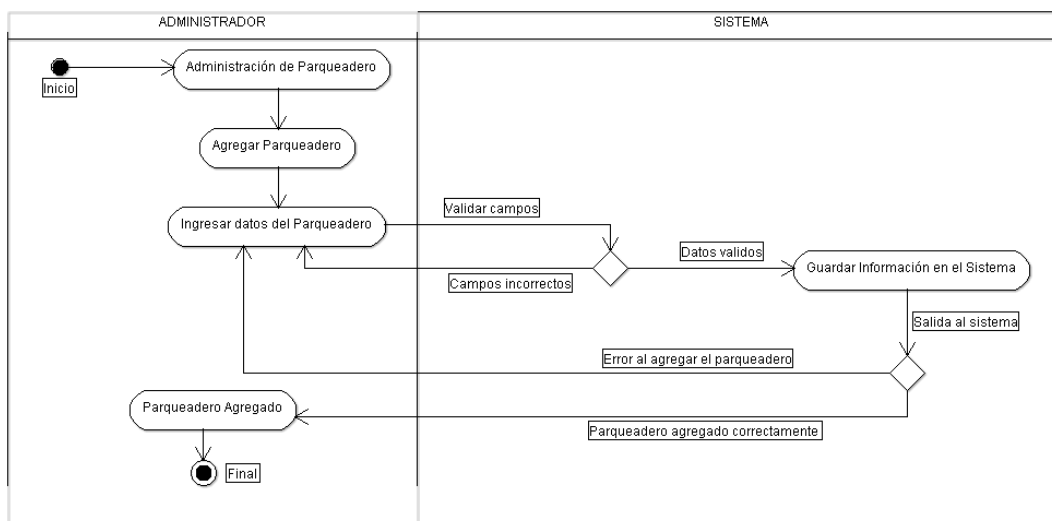
En el siguiente diagrama se muestran las actividades que realiza el usuario para ingresar a la Apps Web y conocer la información y ubicación de los parqueaderos.



**Ilustración 56:** Diagrama de Actividad 02 Sistema GPS

### 5.4.3. DIAGRAMA DE ACTIVIDAD DE ADMINISTRACIÓN DE PARQUEADEROS

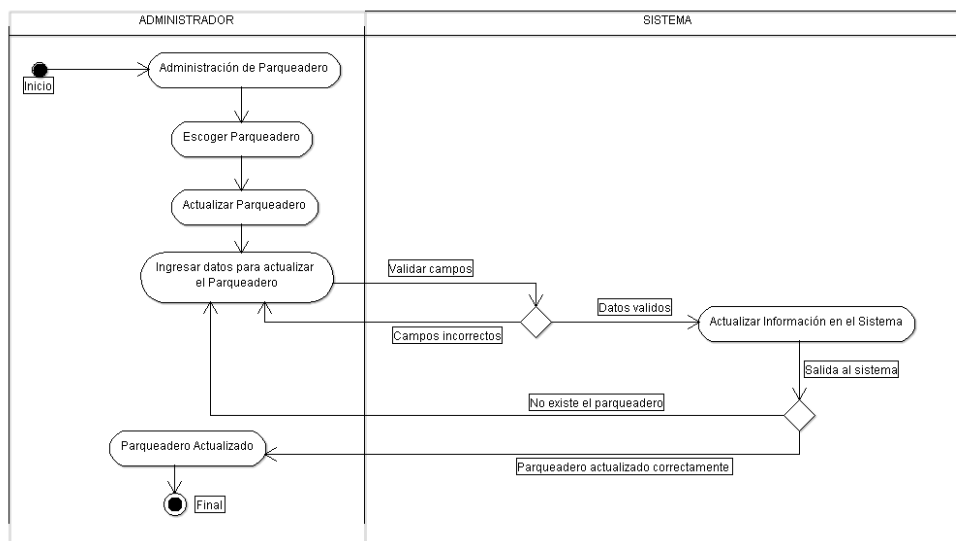
- **Administración de parqueaderos opción agregar:** En este diagrama se describen las actividades que realiza el administrador de la Apps Web en la opción agregar que le permite adicionar parqueaderos al sistema.





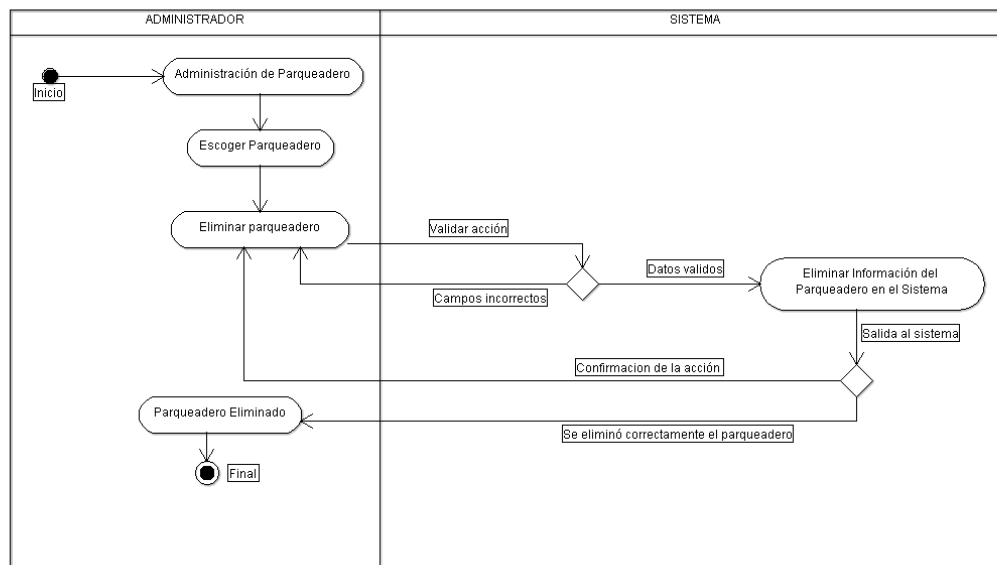
**Ilustración 57:** Diagrama de Actividad 03 Administración de parqueaderos opción agregar.

- **Administración de parqueaderos opción modificar:** En este diagrama se describen las actividades que realiza el administrador de la Apps Web para modificar un parqueadero específico, utilizando un formulario donde se modifican los datos que sean necesarios.



**Ilustración 58:** Diagrama de Actividad 03 Administración de parqueaderos opción modificar.

- **Administración de parqueaderos opción eliminar:** En este diagrama se describe las actividades que realiza el administrador de la Apps Web a la hora de eliminar un parqueadero del sistema a través de una selección que devolverá la información que se quiere eliminar si este existe.

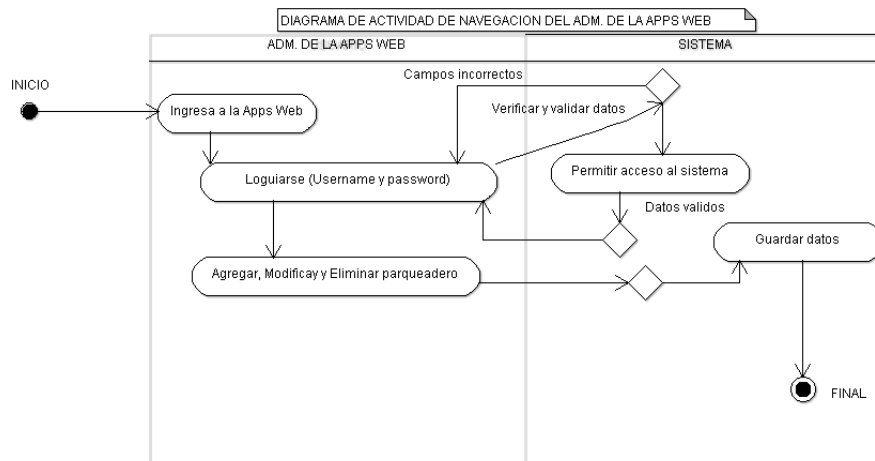


**Ilustración 59:** Diagrama de Actividad 03 Administración de parquederos opción eliminar.

#### 5.4.4. DIAGRAMA DE ACTIVIDAD DE NAVEGACION

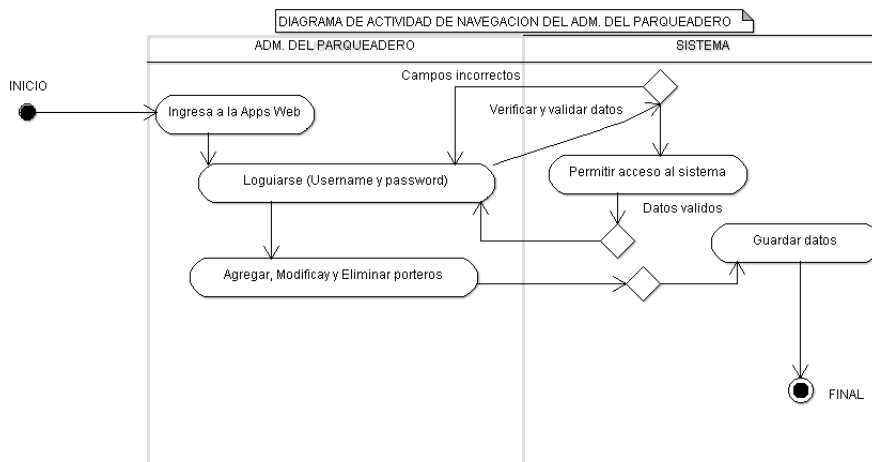
En los siguientes diagramas se describirán las actividades de cada actor según su rol cuando realiza sus funciones interactuando con la base de datos del sistema.

➤ **DIAGRAMA DE ACTIVIDAD DE NAVEGACION DEL ADM. DE LA APPS WEB**



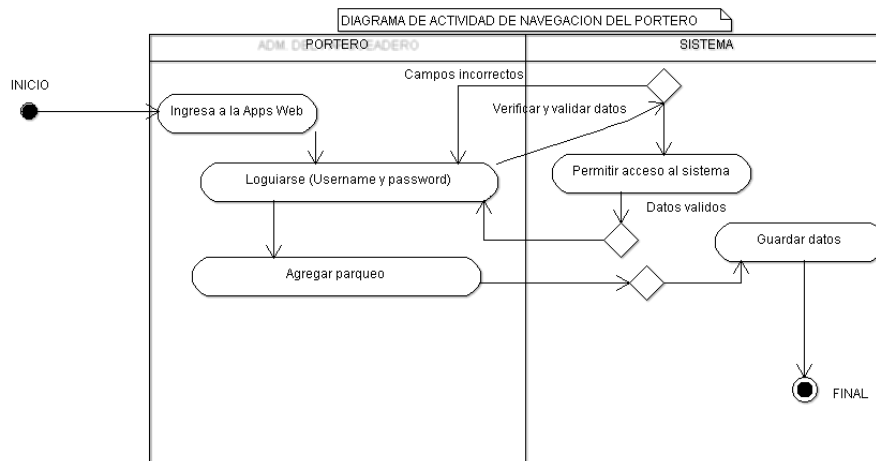
**Ilustración 60:** Diagrama de Actividad 04 Navegación del Administrador de la Apps Web.

➤ **DIAGRAMA DE ACTIVIDAD DE NAVEGACION DEL ADM. DEL PARQUEADERO**



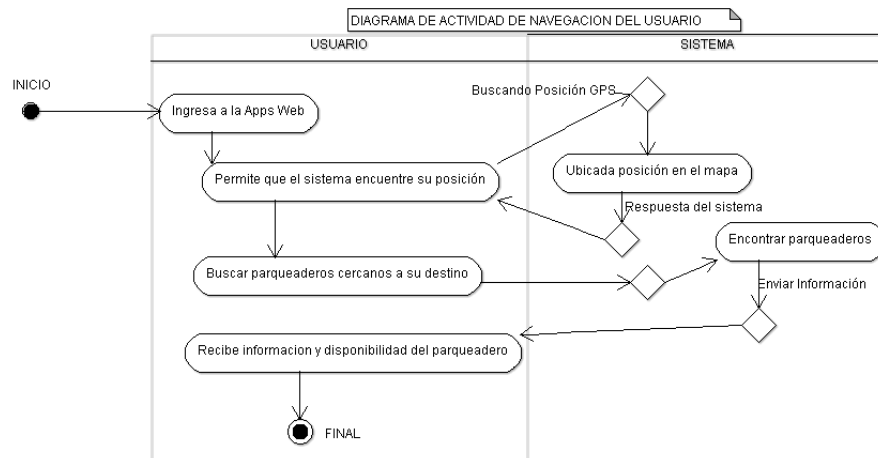
**Ilustración 61:** Diagrama de Actividad 04 Navegación del Administrador del parqueadero.

## ➤ DIAGRAMA DE ACTIVIDAD DE NAVEGACION DEL PORTERO



**Ilustración 62:** Diagrama de Actividad 04 Navegación del portero

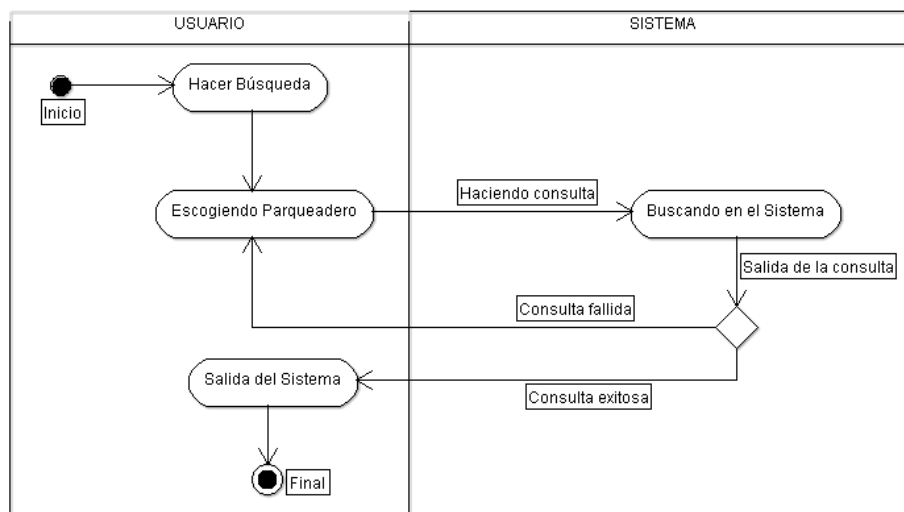
## ➤ DIAGRAMA DE ACTIVIDAD DE NAVEGACION DEL USUARIO



**Ilustración 63:** Diagrama de Actividad 04 Navegación del Usuario

#### 5.4.5. DIAGRAMA DE ACTIVIDAD DE BÚSQUEDA

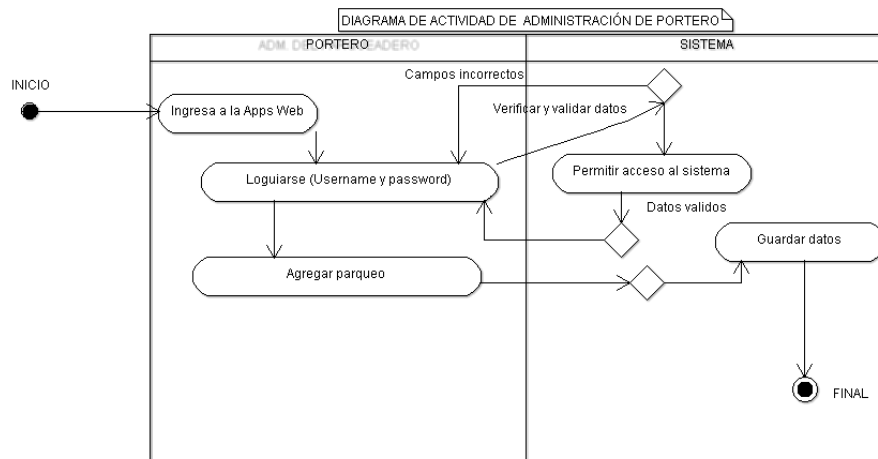
El siguiente diagrama describe las actividades del usuario del sistema cuando realiza sus búsquedas de los productos interactuando con la base de datos del sistema Tienda Virtual.



**Ilustración 64:** Diagrama de Actividad 03 Búsqueda.

#### 5.4.6. DIAGRAMA DE ACTIVIDAD PARA LA ADIMINISTRACIÓN DE PORTERO

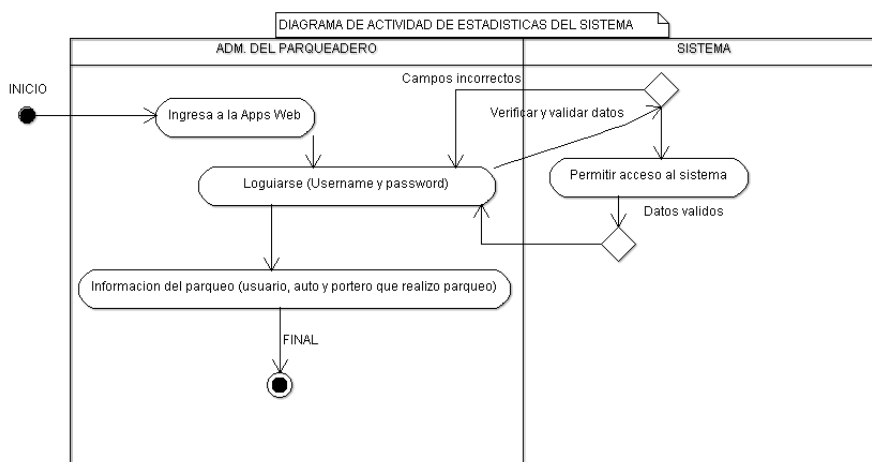
Los siguientes diagramas describen las actividades del portero del parqueadero



**Ilustración 65:** Diagrama de Actividad 04 Administración de usuario opción actualizar.

#### 5.4.7. DIAGRAMA DE ACTIVIDAD PARA LA ESTADISTICA DEL SISTEMA

En el siguiente diagrama se describe la información de todos los parqueos realizados y quien los realizo con fecha y hora.



**Ilustración 66:** Diagrama de Actividad 04 Administración de usuario opción actualizar.

## **5.4.DIAGRAMA DE CLASES**

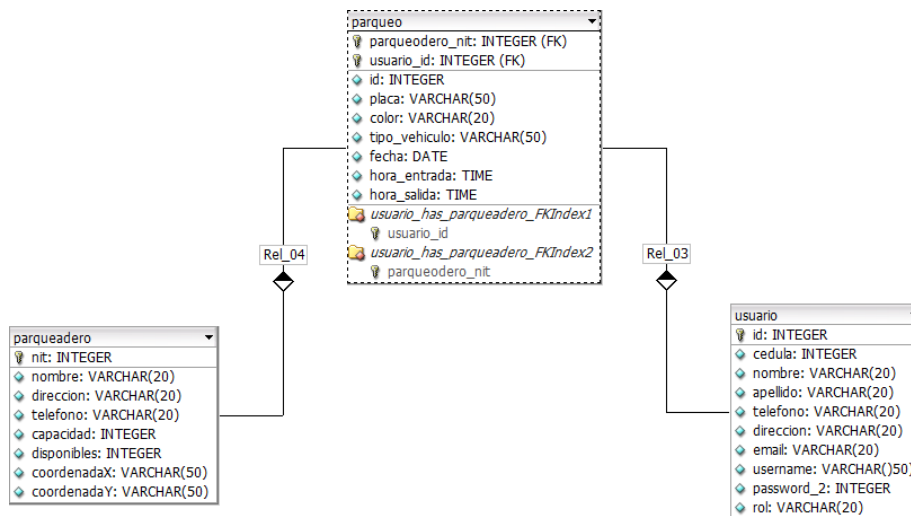
Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos.

El propósito de este diagrama es el de representar los objetos fundamentales del sistema, es decir los que objetos fundamentales del sistema, es decir los que percibe el usuario y con los que espera tratar para percibe el usuario y con los que espera tratar para completar su tarea en vez de objetos del sistema o de un completar su tarea en vez de objetos del sistema o de un modelo de programación.

Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargan del funcionamiento y la relación entre uno y otro.

Representación de requerimientos en entidades y actuaciones. La arquitectura conceptual de un dominio, soluciones de diseño en una arquitectura y componentes de software orientados a objetos.

En el siguiente diagrama veremos las clases y sus respectivas relaciones que nos ayudan a dividir los problemas del sistema Easy Parking en pequeñas partes cada una con funciones fundamentales para que al momento de integrar dichas partes estas nos conlleven a la solución de problemas en el sistema ayudando que los requisitos se satisfagan de manera correcta.



**Ilustración 67:** Diagrama de Clase.

## 5.5. DOCUMENTACIÓN DE CLASES

La siguiente documentación expresa claramente cómo interactúan las clases y las relaciones que existen entre los subsistemas que hacen parte del modelo del sistema Easy Parking.

- **Clase Parqueadero:** En esta clase encontramos los datos de los parqueaderos que se encuentran guardados en el sistema y que prestaran el servicio de parqueo rápido gracias a la Apps Webs.

PARQUEADERO	
RESPONSABILIDAD	COLABORACIÓN
Define los recursos con los que cuenta el sistema Easy Parking para mantener informado a los usuarios.	

**Tabla 20:** Documentación de clase Parqueadero



- **Clase Usuario:** Esta clase representa las relaciones y las responsabilidades que tienen los usuarios dentro del sistema y toda la funcionalidad de los diferentes usuarios que interactúan en el sistema.

USUARIO	
RESPONSABILIDAD	COLABORACIÓN
Define los datos de ingreso (usuario y contraseña) de un usuario del sistema	
Define sus datos personales y adicionales para crear su roll dentro del sistema Tienda Virtual.	

**Tabla 21:** Documentación de clase usuarios.

- **Clase Parqueo:** Esta clase representa el parqueo que se realiza, en esta se encuentran todos los datos relacionados al parqueadero y al usuario que lo registra además de fecha y hora.

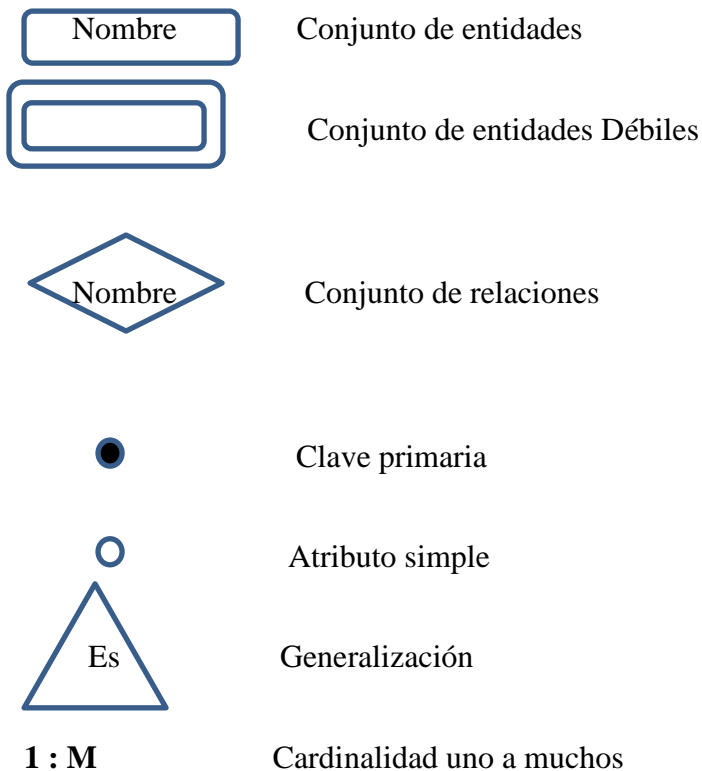
PARQUEO	
RESPONSABILIDAD	COLABORACIÓN
Define el parqueo y toda la información acorde a este.	

**Tabla 22:** Documentación de clase Parqueo.

## 5.6.MODELO ENTIDAD RELACIÓN

El modelo de datos entidad–relación (E-R) está basado en una percepción del mundo real consistente en objetos básicos llamados entidades y de relaciones entre dos objetos. Se desarrolló para facilitar el diseño de bases de datos permitiendo la especificación de un esquema de la empresa que representa la estructura lógica completa de una base de datos. El modelo de datos E-R es uno de los diferentes modelos de datos semánticos; el aspecto semántico del modelo extremadamente útil para hacer corresponder los significados e interacciones de las empresas del mundo real con un esquema conceptual. Debido a esta utilidad, muchas herramientas de diseño de bases de datos se basan en los conceptos del modelo E-R.

### 5.6.1. NOTACIONES UTILIZADAS EN EL MODELO ENTIDAD RELACIÓN

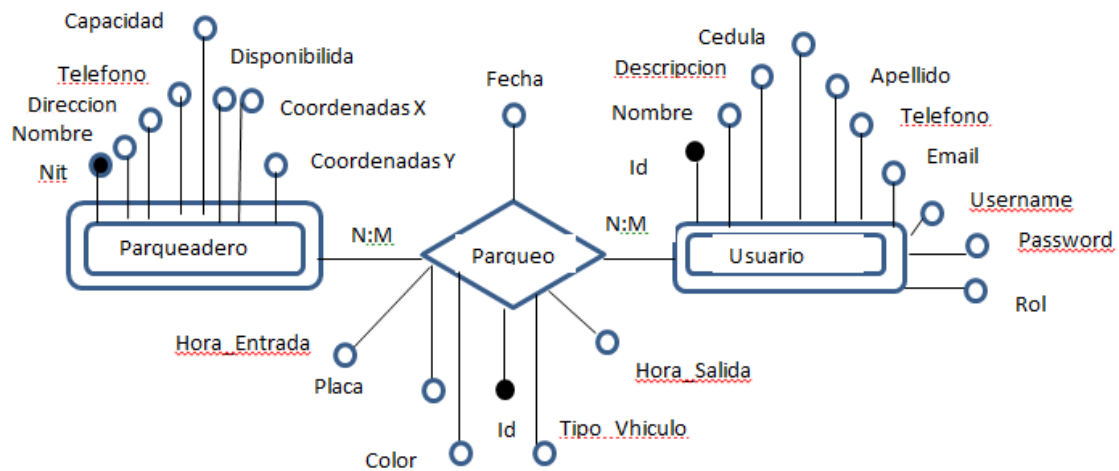


**N : M** Cardinalidad muchos a muchos

**1 : 1** Cardinalidad de uno a uno

 Atributo Multivalorado

## DIAGRAMA ENTIDAD RELACION



**Ilustración 68:** Diagrama Entidad relación